

DEVELOPMENT OF THE DAQ FRONT-END  
FOR THE DSSC DETECTOR AT THE  
EUROPEAN XFEL

INAUGURALDISSERTATION  
ZUR ERLANGUNG DES AKADEMISCHEN GRADES  
EINES DOKTORS DER NATURWISSENSCHAFTEN  
DER UNIVERSITÄT MANNHEIM

VORGELEGT VON  
**Dipl.-Phys. Thomas Gerlach**  
GEBOREN IN HEIDELBERG, DEUTSCHLAND

MANNHEIM, IM MÄRZ 2013

Dekan: Prof. Dr. Heinz Jürgen Müller, Universität Mannheim  
Referent: Prof. Dr. Reinhard Männer, Universität Mannheim  
Korreferent: Prof. Dr. Wolfgang Effelsberg, Universität Mannheim  
Tag der mündlichen Prüfung: 4. Juni 2013



## Entwicklung des DAQ-Frontends für den DSSC Detektor am Europäischen XFEL

Der Europäische XFEL ist eine internationale Photonenforschungseinrichtung, welche derzeit am DESY in Hamburg aufgebaut wird. Ihre einzigartigen Eigenschaften werden neue Möglichkeiten zur Untersuchung kleinster Strukturen, ultraschneller Prozesse und von Materie unter extremen Bedingungen eröffnen. Die Forschung wird unschätzbare Erkenntnisse in Wissenschaftszweigen wie Biologie, Medizin und Chemie, aber auch Nanotechnologie, Astrophysik und anderen liefern. Der DSSC Detektor ist einer von drei 2d-Megapixel-Detektoren, welche derzeit für die Anwendung am XFEL entwickelt werden. Eine Herausforderung stellt das Erfassen der riesigen Mengen an Detektordaten dar. Die geschätzte Nutzdatenrate des DSSC liegt bei 67.2 Gb/s. Diese Arbeit stellt das DAQ-Frontend des DSSC-Systems vor. Ein besonderes Augenmerk liegt auf der Entwicklung des I/O Boards, welches das Basismodul der unteren DAQ-Ebene darstellt. Das DAQ-System des DSSC nutzt modernste Technologien aus Mikroelektronik und Hochgeschwindigkeits-Datenübertragung. Konzipiert als zweistufiges, hierarchisches System, besteht es aus insgesamt 20 Auslesemodulen, welche auf der FPGA-Technologie basieren. Die 16 Basismodule der ersten DAQ-Ebene empfangen Daten vom Detektorfrontend über 256 elektronische Kanäle mit einer gesamten Linkbandbreite von 89.6 Gb/s. Die gesammelten Daten werden dann in vier 3.125 Gb/s Hochgeschwindigkeitsverbindungen pro Modul zusammengeführt und an die vier Hauptmodule der zweiten DAQ-Ebene, den Patch Panel Transceivern, übertragen. Speziell für die Untermodule entwickelte FPGA-Firmware implementiert die Ausleselogik und den Mechanismus zur Bündelung der gesammelten Daten. Zusätzliche Kontrollereinheiten der Firmware sind verantwortlich für das Steuern kritischer Detektorelektronik. Die Testergebnisse und Messungen zeigen, dass das I/O Board sowohl die Datenerfassung bei gegebener Bandbreite als auch die systemnahen Steueraufgaben, die zum korrekten Betrieb des Detektors notwendig sind, in geeigneter Weise durchführt.

## Development of the DAQ Front-end for the DSSC Detector at the European XFEL

The European XFEL is an international photon science facility currently under construction at DESY, Hamburg. Its unique characteristics will open up new research opportunities for investigating tiny structures, ultra-fast processes, and also matter under extreme conditions. The research will allow invaluable insights for many scientific disciplines like biology, medicine, and chemistry, but also for nano-technology, astro-physics, and others. The DSSC detector is one of three 2d megapixel detectors presently being developed for application at the XFEL facility. A challenge is the acquisition of the huge data amount produced by the detector system. The total payload data rate is estimated to be in the order of 67.2 Gb/s. This thesis presents the DAQ front-end for the DSSC detector. A special focus is on the development of the I/O Board, which represents the basic component of the lower DAQ layer. The DSSC front-end DAQ system exploits the features of latest technology in microelectronics and high-speed data transmission. Organized as a two-staged hierarchical system, it comprises 20 readout nodes in total, based on FPGA technology. The 16 slave nodes of the first DAQ layer receive data from the detector front-end at an aggregate link bandwidth of 89.6 Gb/s via 256 electrical links. The accumulated data are then concentrated into four 3.125 Gb/s high-speed links per node for transmission towards the four master nodes of the second DAQ layer, the Patch Panel Transceivers. Custom-built firmware on the slave node FPGAs implements the readout logic and concentrator mechanism for the acquired detector data. It additionally comprises several controller modules, which are responsible for operating critical detector electronics. The test results and measurements show that the I/O Board is able both to manage data acquisition at the required bandwidth and also to perform low-level controlling tasks as required for proper detector operation.



# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>The European XFEL</b>	<b>11</b>
2.1	Free Electron Lasers . . . . .	11
2.1.1	FEL Physics . . . . .	11
2.2	The XFEL Facility at DESY . . . . .	14
2.2.1	Beamline and User Experiments . . . . .	15
2.2.2	Electron Bunch Timing . . . . .	18
2.2.3	Detectors . . . . .	19
2.2.4	Data Handling . . . . .	20
<b>3</b>	<b>The DSSC Detector</b>	<b>23</b>
3.1	The DEPFET Sensor . . . . .	23
3.1.1	Functional Principle . . . . .	23
3.1.2	DEPFET with Intrinsic Signal Compression . . . . .	24
3.2	The DSSC Detector . . . . .	24
3.2.1	Detector Concept . . . . .	26
3.3	Control and Readout Architecture . . . . .	28
3.3.1	Back-end DAQ . . . . .	28
3.3.2	DSSC Front-end DAQ . . . . .	34
3.3.3	Readout ASICs . . . . .	35
<b>4</b>	<b>FPGA-based DAQ Systems in Modern Physics Experiments</b>	<b>37</b>
4.1	DAQ Systems at CERN . . . . .	37
4.2	DAQ Systems for the 2d Detectors at the European XFEL . . . . .	38
4.2.1	The DAQ at AGIPD . . . . .	39
4.2.2	The DAQ at the LPD . . . . .	40
<b>5</b>	<b>The DSSC Front-end DAQ</b>	<b>43</b>
5.1	The DAQ at the DSSC . . . . .	43
5.1.1	The Concept . . . . .	43
5.1.2	Timing and Control of the DSSC DAQ . . . . .	45
5.1.3	Data Transport . . . . .	49
5.2	Hardware Implementation . . . . .	50
5.2.1	The Patch Panel Transceiver . . . . .	51
5.2.2	The I/O Board . . . . .	54
5.3	Connectivity of the DSSC DAQ . . . . .	56
5.4	Comparison of the Front-end DAQ Systems . . . . .	56

<b>6</b>	<b>The I/O Board Prototype</b>	<b>59</b>
6.1	Design Considerations . . . . .	60
6.1.1	Signal Integrity . . . . .	60
6.1.2	Interfaces and Signals to External Electronics . . . . .	63
6.1.3	FPGA I/O Assignment . . . . .	64
6.2	PCB Layout . . . . .	65
6.2.1	PCB Layer Stack-up . . . . .	67
6.2.2	Split Power and Ground Planes . . . . .	68
6.2.3	Impedance Calculation . . . . .	68
6.2.4	Length Matching . . . . .	69
6.2.5	Signal Termination . . . . .	70
6.2.6	Power Bypassing . . . . .	70
6.3	Special Circuitries . . . . .	70
6.3.1	Local Power Supplies . . . . .	70
6.3.2	Sensor Power Switching . . . . .	71
6.3.3	Sensor Clear Signals . . . . .	74
6.4	I/O Board Rev. 1.0 . . . . .	75
<b>7</b>	<b>The I/O Board FPGA Firmware</b>	<b>77</b>
7.1	VHDL Firmware . . . . .	77
7.1.1	Clocking Structure and Local Timing . . . . .	78
7.1.2	System Configuration and Control . . . . .	80
7.1.3	Peripheral Control . . . . .	84
7.1.4	ASIC Data Transport . . . . .	93
7.2	Simulation and Verification . . . . .	97
7.3	Test Environment . . . . .	97
7.3.1	The MPRACE-2 Board . . . . .	98
7.3.2	MPRACE-2 FPGA Firmware . . . . .	100
7.3.3	MPRACE-2 FPGA Software . . . . .	101
<b>8</b>	<b>Signal Analysis and System Verification</b>	<b>103</b>
8.1	Slow-control Interface . . . . .	103
8.2	ASIC Control Interface . . . . .	104
8.3	ASIC Data Readout . . . . .	104
8.4	High-speed Transceivers . . . . .	107
8.5	PRB Control Interface . . . . .	109
8.6	Main Board Clock Buffer Interface . . . . .	111
8.7	FET Driver Control Interface . . . . .	114
8.8	Sensor Clear Signal Interface . . . . .	116
8.9	Summary of Test Results . . . . .	118
<b>9</b>	<b>Conclusion and Outlook</b>	<b>121</b>
	<b>Appendix</b>	<b>125</b>
<b>A</b>	<b>I/O Board</b>	<b>127</b>
A.1	FPGA Power Estimation . . . . .	127
A.2	Signal Naming . . . . .	128

<b>B</b>	<b>VHDL Firmware Modules</b>	<b>129</b>
B.1	SysConfig . . . . .	129
B.1.1	Entity Port Declaration . . . . .	129
B.2	SysCtrl . . . . .	131
B.2.1	Entity Port Declaration . . . . .	131
B.2.2	Simulation . . . . .	131
B.3	LmkCtrl . . . . .	132
B.3.1	Entity Port Declaration . . . . .	132
B.3.2	SyscConfig Register Map . . . . .	132
B.3.3	Simulation . . . . .	132
B.4	PrbCtrl . . . . .	133
B.4.1	Entity Port Declaration . . . . .	133
B.4.2	SysConfig Registers . . . . .	133
B.4.3	Simulation . . . . .	134
B.5	FetCtrl . . . . .	135
B.5.1	Entity Port Declaration . . . . .	135
B.5.2	SysConfig Registers . . . . .	135
B.5.3	Simulation . . . . .	136
B.6	ClrCtrl . . . . .	137
B.6.1	Entity Port Declaration . . . . .	137
B.6.2	SysConfig Registers . . . . .	137
B.6.3	Simulation . . . . .	138
B.7	AsicRoCtrl . . . . .	139
B.7.1	Entity Port Declaration . . . . .	139
B.7.2	Simulation . . . . .	139
B.8	MPRACE-2 SysConfig Map . . . . .	140

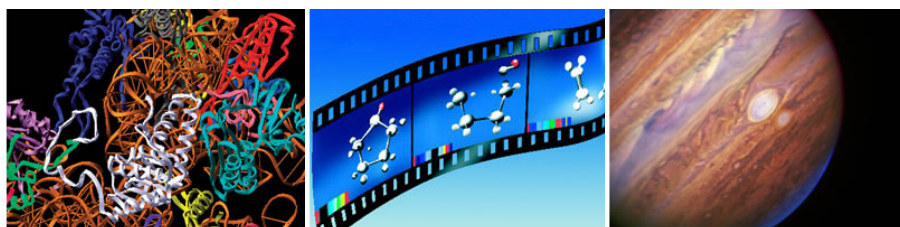


# 1 Introduction

The ambition for understanding the fundamental processes and structure of matter has always been a motivation for scientists to develop new research facilities that provide even better techniques for investigation. While high-energy experiments applied in particle physics – like the LHC at CERN – study the characteristics of matter on a sub-atomic level, *photon science facilities* use light sources for research on the properties of matter on a molecular scale. Of particular interest are light sources that emit *coherent* light as provided by lasers. The coherence and intensity of the light allows analyzing the structure of a sample by drawing conclusions from the recorded patterns that evolve from scattering and diffraction.

The spectrum of conventional lasers usually is in the range of a few hundred nanometers (700 nm visible down to 150 nm ultra-violet). In order to resolve structures on a molecular scale, it is necessary to use sources of light with even smaller wavelengths. On the other hand, the time scale of many ultra-fast processes is in the order of less than a trillionth of a second (picosecond,  $10^{-12}$  s). Recording snapshots of intermediate states during such processes requires laser pulse widths with magnitudes in the femtosecond range, which is not provided by conventional lasers.

The European XFEL at the DESY in Hamburg, Germany, is one of the recent activities in the development of photon science facilities. The free electron laser will provide pulses of full coherent



**Figure 1.1:** The European XFEL will provide benefits to the investigation of tiny structures, ultra-fast processes, and matter under extreme conditions. Source: [1]

X-rays with wavelengths in the (sub-) nanometer range ( $0.05 \text{ nm} < \lambda < 4.7 \text{ nm}$ ) that allow resolving tiny structures beyond molecular scale. With a pulse width of less than 100 fs and a temporal spacing of 220 ns, the time scale of the light flashes is of particular importance, as atoms and molecules oscillate around their equilibrium positions with periods of a few hundred femtoseconds. Light sources with such pulse characteristics allow for studying intermediate states of ultra-fast processes on a femtosecond time scale. Furthermore, due to the coherence and the extremely high intensity, the X-ray pulses will provide unique scattering and diffraction patterns that will help to analyze and understand complex molecular structures.

The European XFEL aims for a variety of experiments in different scientific research areas. Deciphering of biomolecules and the three-dimensional exploration of nano structures are examples for the investigation of tiny structures. The delivered X-ray pulses are so intense that single molecules can be studied – present X-ray sources are too weak and rely on the use of molecule crystals instead,

of which many pictures are then superposed. Also, due to the very short duration of the flashes, the samples hardly change during the exposure, but decay afterwards.

The study of ultra-fast processes involves capturing intermediate states and pathways of chemical reactions and phase transitions. The uniquely short duration of an X-ray pulse allows snapshots of moving details to be recorded without becoming blurred. The provided range of wavelengths simultaneously makes atomic details visible. An example would be the investigation of magnetization to understand the processes that occur when materials reverse their magnetic polarization.

Finally, the observation of small objects in strong fields is an example for research on matter under extreme states. The intense X-ray radiation field creates a unique environment in which the investigated samples are excited into highly-ionized states, which were unable to observe so far.

The DEPFET Sensor with Signal Compression (DSSC) is one of three 2d megapixel detector systems currently being developed for application at the European XFEL to record the diffraction patterns produced by the laser beam scattered on a sample. The detector is based on an enhanced version of the DEPFET sensor technology. Its non-linear signal characteristics allows for a high dynamic range to handle large amounts of photons per pixel, while simultaneously providing high sensitivity at low signal charges for single photon resolution. The DSSC is designed for photon energies between 0.5 keV and 20 keV ( $\lambda_{ph} = 2.4 \text{ nm} \dots 0.062 \text{ nm}$ ), which makes it suitable for a variety of experiments applied at the XFEL facility.

A major challenge in the development of the 2d detectors is the transport of the huge amount of data produced by the sensor electronics due to the challenging X-ray pulse timing. About 2,700 pulses are delivered within 600  $\mu\text{s}$ , followed by a readout phase of 99.4 ms duration, before a new macro bunch of flashes is provided. Current technology limits the detectors to buffer only a fraction of a few hundred images per macro-bunch. However, the expected data rates are still at a level which demands for a high-performance Data Acquisition (DAQ) system. Regarding the DSSC, the detector is capable of storing 640 frames per macro-bunch, and the estimated payload output data rate for the entire system is in the order of 67.2 Gb/s.

Designing a suitable DAQ system for the DSSC detector also depends on other important aspects. System-specific requirements – like complying with operating conditions, electrical interfaces, and mechanical constraints – can hardly be realized with commercially available standard hardware. Moreover, the local detector DAQ must be compatible with a central back-end DAQ system which is common to all 2d detectors applied at the XFEL facility. The back-end system raises special demands on both the electrical interfaces and the packet format for data transmission, as it expects pixel-ordered frame data. Especially the latter accounts for pre-processing the recorded detector data during the readout phase. Minimum latency is thus an important prerequisite the DSSC DAQ must provide, as only a fraction of the detector dead time can be used for processing and transmitting the data. Hence, the development of custom hardware is inevitable for the implementation of an adequate DSSC DAQ system.

**Outline** This thesis focuses on the development of the first layer of the front-end DAQ system for the DSSC detector at the European XFEL. The following chapter 2 introduces the European XFEL facility, including its experiment stations and detector systems. It additionally gives an overview of the fundamentals of free electron laser physics. In chapter 3, the general concept of the DSSC detector is presented. A deeper insight into the functional principle of the DEPFET sensor is given as well as details on the readout and control architecture of the system. Chapter 4 introduces the advantages of FPGA-based DAQ systems as applied in many modern physics experiments and also presents the DAQ systems of the other detectors at the XFEL facility. The concept of the DSSC DAQ system is explained in chapter 5, followed by an elaborate description of its



---

FPGA-based hardware implementation. The chapter concludes with a comparison of the different XFEL DAQ systems. In chapter 6, the hardware architecture of one of the DAQ modules is presented, which is designed to provide both low-level control of detector electronics and high-speed readout of sensor data. Both general design considerations and special hardware features are introduced. Chapter 7 elaborately describes the FPGA firmware developed for the DAQ node presented in the previous chapter. It additionally pictures the environment built to extensively test and verify both electrical and logical functionality of the hardware and firmware. The acquired results of measurements on signal transmission characteristics and analysis of firmware functionality are summarized in chapter 8. Chapter 9 eventually summarizes the achievements of the development work and concludes with future prospects of the DSSC DAQ system.



## 2 The European XFEL

At the Deutsches Elektron Synchrotron (DESY), the German research center for particle physics in Hamburg, scientists develop and operate particle accelerators to study the fundamental characteristics and structure of matter. Founded in 1949, DESY specializes in particle physics and photon science and has provided its resources and research facilities to over 3,000 scientists of more than 40 nations.

One of the most recent activities in the development of particle accelerators and photon science facilities at DESY is the European X-ray Free Electron Laser (XFEL), an international X-ray laser research project currently under construction. The facility is intended to be operational<sup>1</sup> by 2015, and will produce ultra-short X-ray flashes of highest brilliance. The intense laser flashes will enable scientists to map atomic details of tiny structures, film ultra-fast processes, and study extreme states of matter.

The first part of this chapter provides an introduction to the fundamental physics and concept of free electron lasers in general. The second part addresses the XFEL facility at DESY. In particular, a brief description of its experiments and detectors is given.

### 2.1 Free Electron Lasers

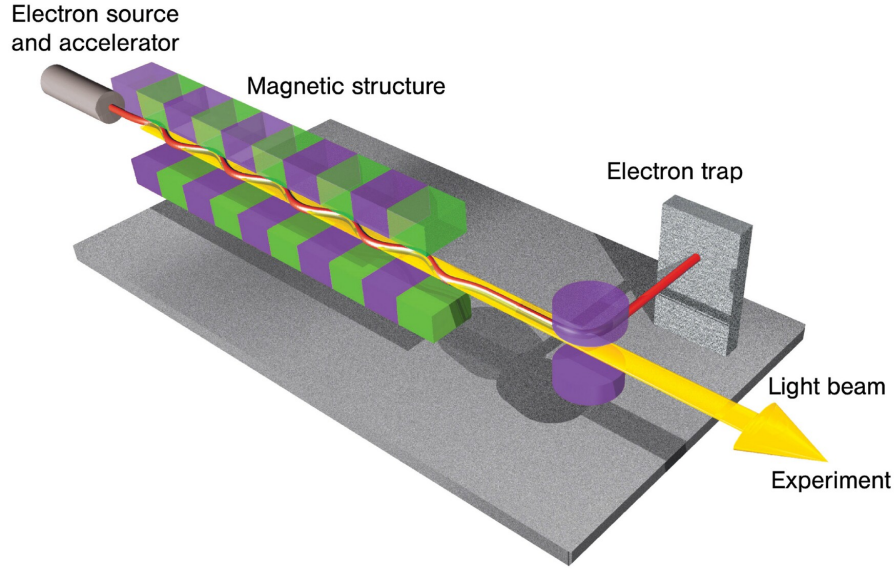
In its basic concept, a Free Electron Laser (FEL) is a synchrotron source emitting coherent *synchrotron radiation*. Unlike a conventional laser, where the amplification is a result of the stimulated photon emission of electrons bound to atoms, the radiation of a free electron laser is emitted by free (unbound) electrons travelling through a magnetic field at relativistic velocity. Therefore, the fundamental principle of an FEL can be described within the theory for generating synchrotron radiation.

#### 2.1.1 FEL Physics

The radiation produced by electrons of conventional synchrotron facilities is emitted incoherently [2, 3]. However, the research of Nakazato et al. [4] and Blum [5] show that *coherent* synchrotron radiation is emitted if the electrons are spatially bunched on the scale of the radiation wavelength. In an FEL, this is accomplished with a special, periodic arrangement of magnetic dipoles, a so-called *undulator*, which forces the electrons to travel on a sinusoidal trajectory. The resonant interaction of the irradiated electro-magnetic field with the electrons causes them to form small bunches, and exponentially increases the intensity of the radiation. Figure 2.1 illustrates the basic principle of an FEL. Since the electrons are not bound to atoms, and thus are not limited to specific transitions between two discrete energy levels, the wavelength of an FEL is tunable over a wide range. It depends only on the energy of the accelerator and on parameters of the undulator magnet.

---

<sup>1</sup>As of 2012, cf. <http://www.xfel.eu/project>



**Figure 2.1:** Principle of an FEL. A linear accelerator speeds up electrons to relativistic velocity. The electron beam (red) is then directed through an undulator magnet (green/violet), which causes the electrons to travel on a sinusoidal trajectory and emit coherent synchrotron radiation in forward direction (yellow). Copyright European XFEL. Source: [6]

## Synchrotron Radiation and Undulator Radiation

The following summary of fundamental FEL physics is mainly based on the considerations given in [7].

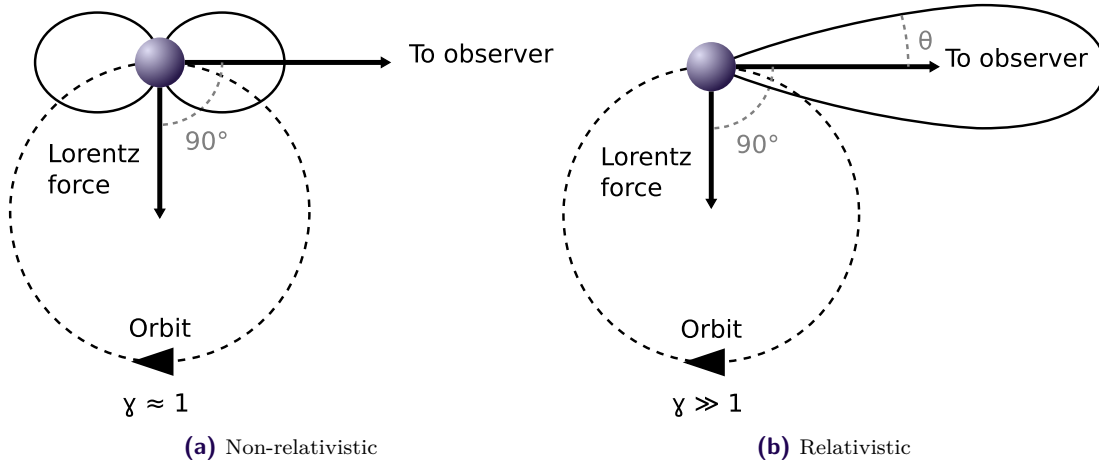
An electron travelling through a magnetic field is accelerated perpendicular to its trajectory due to the introduced Lorentz force. At relativistic energies, the electron will emit electromagnetic radiation known as synchrotron radiation, as illustrated in figure 2.2. For the angle  $\theta$  of radiation applies

$$\tan(\theta) = 1/\gamma \quad (2.1)$$

in which  $\gamma = E/(m_e c^2)$  is the Lorentz factor<sup>2</sup>, and  $E$  is the electron energy. The electron rest mass is given by  $m_e = 512 \text{ keV}/c^2$ , and  $c$  is the speed of light (in vacuum). That is, the radiation is emitted almost tangentially to the trajectory of electrons at relativistic energies ( $\gamma \gg 1$ ). When relativistic electrons propagate through an undulator, the Lorentz force of the alternating magnetic fields causes the charges to travel on a transversal trajectory, and synchrotron radiation is emitted in forward direction. The deflection of the electrons from the forward direction is comparable to the opening angle  $\theta$  of the synchrotron radiation cone.

**Resonance condition** As the electrons traverse the individual magnetic periods, additional radiation wavefronts are emitted, which interfere with each other. This interference effect can be

<sup>2</sup>The Lorentz factor describes the reciprocal of the length contraction and the time dilatation within the theory of relativity. It is named after H. Lorentz, a Dutch physicist.



**Figure 2.2:** Generation of synchrotron radiation. At relativistic velocities ( $\gamma \gg 1$ ), synchrotron radiation is emitted almost tangentially in forward direction of the electron trajectory.

formulated as

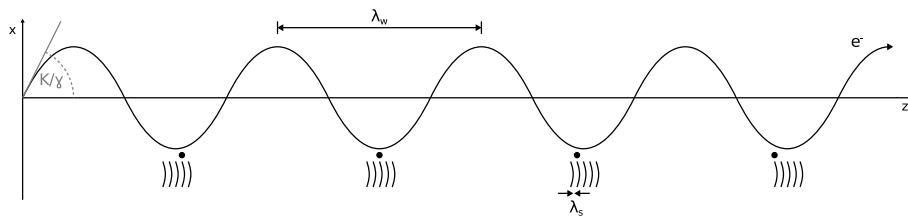
$$\lambda_{ph} = \frac{\lambda_u}{2\gamma^2} (1 + K_{rms}^2) \quad (2.2)$$

in which  $\lambda_{ph}$  is the wavelength of the first harmonic of the spontaneous on-axis undulator emission, and  $\lambda_u$  is the length of the magnetic period of the undulator. The averaged undulator parameter

$$K_{rms} = \frac{eB_u\lambda_u}{2\pi m_e c} \quad (2.3)$$

represents the ratio between the average deflection angle of the electrons and the typical opening angle of the synchrotron radiation cone.  $B_u$  is the rms<sup>3</sup> magnetic field of the undulator and  $e$  is the electron charge.

The resonance condition requires that with each undulator period traversed, the electron slips by one radiation wavelength with respect to the (faster) electromagnetic wave, as shown in figure 2.3.



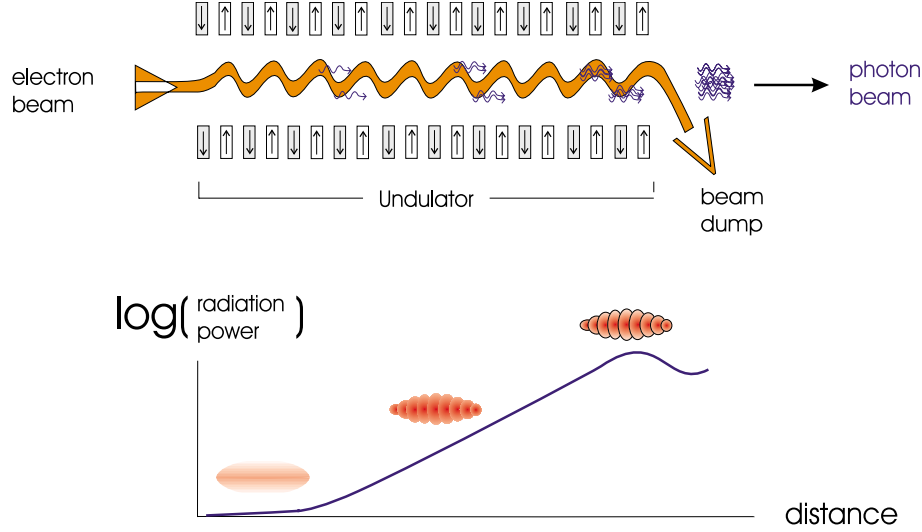
**Figure 2.3:** Resonance condition in an undulator. Source: [8]

## The SASE Principle

Self-amplified Spontaneous Emission (SASE) refers to the amplifying process that occurs when electrons oscillating through an undulator interact with the synchrotron radiation they emit.

<sup>3</sup>Root mean square

Depending on the relative phase relation between the electron oscillation and the electromagnetic wave, the electrons are either accelerated or decelerated. A phase difference of  $\Delta\phi = 0^\circ$  reduces the energy of the electrons, while a phase difference of  $\Delta\phi = 180^\circ$  increases it. Consequently, the longitudinal distribution of electrons in the bunch changes so that equidistant slices evolve, with a spatial separation that corresponds to the wavelength  $\lambda_{ph}$  of the emitted radiation. This process is called *micro-bunching* and is illustrated in figure 2.4. Before micro-bunching applies, the electrons



**Figure 2.4:** Illustration of SASE principle (top) and micro-bunching (bottom). Source: [7]

of a bunch can be seen as individually radiating charges without a common phase relation. The power of the spontaneous emission is proportional to the number of electrons  $N_e$  of a bunch. Over time, the number of micro-bunched electrons radiating in phase increases until the micro-bunching has completed. The radiation power is then proportional to  $N_e^2$ , which means an amplification of many orders of magnitude with respect to the spontaneous emission of the undulator.

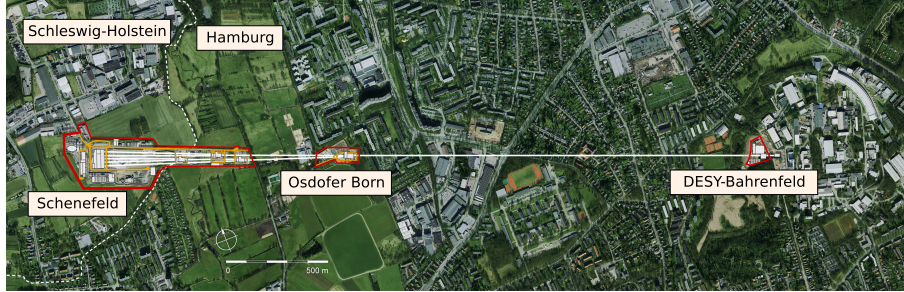
One major prerequisite for SASE is the resonance condition presented above. However, in order to gain exponential amplification of the synchrotron radiation generated in an undulator, an electron beam of excellent quality must be supplied. Additionally, a sufficient overlap between the radiation wavefront and the electron bunch is required. This is accomplished by a low-emittance and low-energy spread electron beam, which additionally features a high charge density. The undulator, on the other hand, must provide a very precise magnetic field over a sufficiently long distance (typically in the order of several meters).

## 2.2 The XFEL Facility at DESY

The free electron laser of the European XFEL facility will provide coherent and ultra-short X-ray pulses generated by the principle of SASE. The laser flashes will have unique characteristics such as pulse duration below 100 fs, wavelengths in a range from 0.05 nm to 4.7 nm, and very high brilliance of about  $10^{33}$  photons/s/mm<sup>2</sup>/mrad<sup>2</sup> per 0.1 % bandwidth [9].

The facility is built in tunnels about 6 m to 38 m underground and has a total length of 3.4 km. Starting at the DESY site, it extends in north-western direction up to the research campus in

Schenefeld, where the experiment halls are located. Figure 2.5 shows an aerial photograph of the construction area, overlain by a schematic drawing of the main tunnels and the three different sites. At Osdofer Born, the electron bunches are distributed to the tunnels in which they will produce X-ray radiation. The employed super-conducting linear accelerator has a total length of 2.4 km



**Figure 2.5:** The construction area of the European XFEL. The tunnel beneath the surface starts at the DESY site in Hamburg-Bahrenfeld and heads north-west, until it reaches the research campus in Schenefeld. The tunnel fan where the electrons are distributed to the SASE stations and undulators begins at Osdofer Born. Copyright European XFEL. Source: [1]

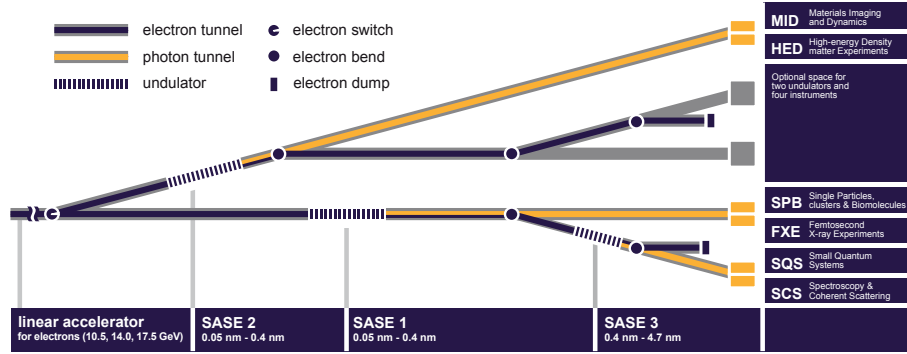
(acceleration length 1.7 km) and provides energies of up to 17.5 GeV. It consists of 101 subsequent cavities (so-called *resonators*) which are cooled down with liquid Helium to a temperature of  $-271^\circ\text{C}$ . Table 2.1 summarizes the key characteristics of the European XFEL facility.

General		
Total length	3.4 km	From DESY site to Schenefeld
Number of sites	3	DESY Bahrenfeld, Osdofer Born, Schenefeld
Tunnel depth	6 m to 38 m	Covered by at least 6 m of soil
Tunnel diameter	up to 5.3 m	
Accelerator		
Type	Super-conducting linear accelerator	
Total length	2.1 km	
Acceleration length	1.7 km	
Max. energy	17.5 GeV	Expandable up to 20 GeV
Temperature	$-271^\circ\text{C}$	Cooled with liquid helium
Number of modules	101	
Properties of X-ray Laser Flashes		
Flashes per second	27,000	
Wavelength	0.05 nm to 4.7 nm	
Duration	$< 100\text{ fs}$	
Brilliance	$5 \cdot 10^{33}$ (peak) $1.6 \cdot 10^{25}$ (average)	$\frac{\text{photons}}{\text{s} \cdot \text{mm}^2 \cdot \text{mrad}^2}$

**Table 2.1:** Summary of the key characteristics of the European XFEL. Source: [9]

## 2.2.1 Beamline and User Experiments

The beamline of the European XFEL is schematically illustrated in figure 2.6. In horizontal



**Figure 2.6:** Beamlines of the European XFEL. Copyright European XFEL. Source: [10]

direction, it can be divided into three sections, the linear accelerator, the three SASE systems and the undulators, and the experiment stations. While the photons produced in SASE stations 1 and 2 are in the lower wavelength regime of 0.05 nm to 0.4 nm, SASE 3 covers the range between 0.4 nm and 4.7 nm. Unlike the photon beams, which aim at the targets in the experiment halls, the electron bunches are terminated in two beam dumps. The major characteristics of the SASE systems and the undulators are listed in table 2.2.

	SASE 1 + 2	SASE 3
Photon wavelength	0.4 nm to < 0.05 nm	4.7 nm to 0.4 nm
Photon energy	3 keV to > 25 keV	0.26 keV to 3 keV
Magnetic length	175 m	105 m
Undulator period length	40 mm	68 mm
Magnetic gap	10 mm to 28 mm	
Undulator segment length	5 m	
Undulator segment gap length	1.1 m	
Instruments	SPB, FXE (SASE 1)	SQS, SCS
	MID, HED (SASE 2)	

**Table 2.2:** Characteristics of the SASE systems and the undulators. Source: [10, 11]

There are six user experiments installed in the underground halls of Schenefeld, which pursue different aims in research. In the following, the experiments are described in some detail.

**SPB** The study of 2d and 3d structures of single particles in the gas phase is the aim of *Single Particles, clusters, and Biomolecules*. The coherent X-rays are scattered on different kinds of particles such as single biomolecules, entire cells, and micro-organisms. Other interesting targets are nano-crystals and atomic clusters. By detecting and analyzing the coherent diffraction pattern it is possible to investigate the structures of the samples. The requirements on the photon energy is in the range of 3 keV to 16 keV [11]. An atomic resolution of better than 1 nm is aimed for the experiments. Scientific areas like material sciences, nano-materials, and structural and cell biology will profit from the results of this experiment.



**FXE** The *Femtosecond X-ray Experiments* investigate ultra-fast processes in condensed-matter systems. The experiment primarily focuses on both liquid and solid systems, but the investigation of gaseous samples is also foreseen. The samples are analyzed using spectroscopy and different kinds of diffraction, like Bragg, powder, and amorphous. Photons of energies between 3 keV and 20 keV will be required [11]. The experiment will contribute to scientific areas like chemical dynamics, dynamics of condensed matter, and matter under extreme conditions.

**MID** The purpose of *Materials Imaging and Dynamics* is the investigation of structure and dynamics of materials on the nano-scale by detecting diffraction patterns produced when the coherent X-rays are scattered on the samples. Typical samples are nano-crystals, nano-structures on or buried under surfaces, or crystallites of a compound sample. A resolution in the 10 nm range is aimed for studying the 2d and 3d structures of condensed-matter samples. The experiment is designed for photon energy ranges from 5 keV to 10 keV and from 25 keV to 36 keV [11]. Scientific areas of application are material sciences, nano-materials and dynamics of condensed matter.

**HED** The *High-Energy Density matter experiments* aim at the research of matter under high-energy density conditions. The energy stored in an X-ray typical volume of  $10 \mu\text{m}^3$  is of about 0.1 mJ or more in this experiment. Plasma physics, planetary physics, and the study of matter under extreme conditions are the scientific scopes. The instrument uses a combination of intense, ultra-short X-ray flashes and high-energy optical laser pulses to produce the HED states in the samples. Inelastic scattering and spectroscopy, as well as various kinds of diffraction like Bragg, powder, and amorphous, are applied for investigation. The demands on the photon energies are in the ranges of 3 keV to 16 keV and 25 keV to 36 keV [11].

**SQS** Processes in atoms, ions, small molecules and clusters, which occur under highly intense beams, will be studied by the *Small Quantum Systems* instrument with the help of different spectroscopy methods. The samples are typically provided in gaseous form, in jets or beams, or by trapping. The research will contribute to scientific areas like atomic and molecular physics, chemical dynamics, optical phenomena, and matter under extreme conditions. The experiment operates with photons at energies in the range of less than 0.28 keV to 3 keV [11].

**SCS** The *Spectroscopy and Coherent Scattering* experiment aims at the investigation of dynamics and electronic and atomic structure of soft matter. But also biological structures and magnetic materials and structures will be studied by this instrument. A resolution in the 10 nm range is targeted for studying the 2d and 3d structures of condensed-matter samples. The study of electronic structures will be enabled by resonant inelastic scattering and emission spectroscopy, while resonant magnetic scattering is used to investigate ultra-fast magnetic processes of materials on the nano-scale. Scientific areas of application are material sciences, structural and cell biology, nanomaterials, and dynamics of condensed matter. The experiment is designed for photons with energies less than 0.28 keV and up to 3 keV [11].

Table 2.3 summarizes the experiments at the European XFEL and their applications.

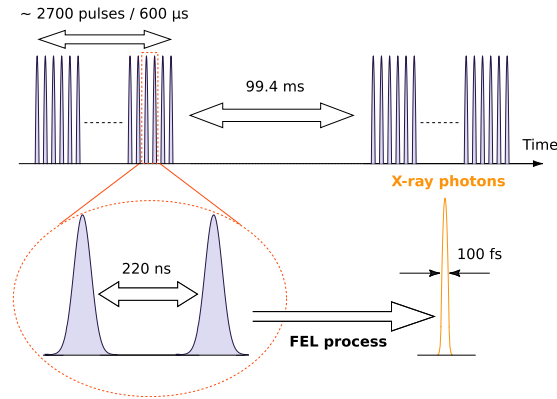
Experiment	Tiny structures	Ultra-fast processes	Extreme states	Light source
SPB	x	x		SASE 1
FXE		x		SASE 1
MID	x	x		SASE 2
HED		x	x	SASE 2
SQS		x	x	SASE 3
SCS	x	x		SASE 3

**Table 2.3:** List of the experiments installed at the European XFEL and their applications.  
Source: [12]

## 2.2.2 Electron Bunch Timing

The coherent X-ray pulses generated by the XFEL machine comply with a very demanding timing, which is illustrated<sup>4</sup> in figure 2.7. There are two relevant pulse structures to distinguish, the *macro-bunches* (in the following also referred to as *trains*) and the *bunches*.

A train is repeated every 100 ms (10 Hz). It is composed of 2,700 bunches equally spaced within 600  $\mu$ s, meaning a temporal distance of 220 ns, and a bunch frequency of 4.5 MHz, respectively. Considering the train period, the inter-train gap yields to 99.4 ms. The FEL process eventually generates the ultra-short X-ray flashes at bunch frequency and a duration of less than 100 fs.



**Figure 2.7:** X-ray bunch timing of the European XFEL. The XFEL machine generates so-called trains with a repetition rate of 10 Hz and a duration of 600  $\mu$ s. A train is composed of 2,700 electron bunches with a temporal distance of 220 ns, which yields an inter-train gap of 99.4 ms. Source: [13]

**Bunch patterns** The XFEL allows for multiple bunch spacing that can be defined individually for each macro-bunch and each beamline. The different bunch spacings are called *bunch patterns* and are stored in a table that defines the intended the number of bunches in an undulator or SASE beamline and the charge range of the bunches. A bunch pattern ID notifies the detector systems of the actual bunch pattern applied.

<sup>4</sup>The image shown is an updated version of the referenced picture, which assumed the initially foreseen pulse frequency of 5 MHz.

### 2.2.3 Detectors

Special pixel detector systems are employed at the European XFEL to record the two-dimensional light patterns that evolve from the scattered X-ray flashes. The patterns allow scientists to gain information about the structure of the sample and the dynamic of a process. The unique qualities of the free electron laser demand for detector systems with sophisticated recording capabilities with regard to time resolution and data output bandwidth.

The large number of pulses per second is one of the major differences between the European XFEL and other XFEL sources available at present. In order to provide single shot imaging, the detectors have to be capable of recording a complete image every 220 ns during the active train period. At the same time, as many images as possible have to be buffered by the detectors for readout during the inter-train gap, since current detector technology is limited and does not allow for transmitting a full image within 220 ns. For data analysis, it is important to know the exact location and count of photons within the sensitive detector area. While the location of photons is determined by the position of the pixel hit in the detector, the count of photons is calculated from the energy deposited inside a pixel. For providing both detection of high count of photons and single photon resolution, the detectors must have a *high dynamic range*. In order to have a reliable image recording system, it is crucial that the detectors are designed *radiation tolerant* to minimize the damage caused by X-rays especially in the high-energy range. Additionally, there are some application-specific demands on the detectors which must be met – like angular coverage and resolution, pixel size, and number of pixels. A detailed elaboration of the detector requirements can be found in [13].

While the requirements described above are primarily related to the sensor electronics, there are also special demands on the DAQ and readout electronics of both a detector and the European XFEL in general to deal with the high amounts of data produced by the detectors.

Three silicon-based megapixel detectors are presently being developed for application at the European XFEL. The systems are designed to meet the requirements and to cover all aspects and specifications of the experiments. In the following, a brief description of the detectors is given. Though different design approaches, the devices share a common feature, the *hybrid pixel technology*. Each pixel of a detector incorporates not only sensor electronics, but also filtering stages and memory cells for buffering the recorded data.

**AGIPD** The development of the *Adaptive Gain Integrating Pixel Detector* is a collaboration between DESY, the University of Bonn (Germany), the University of Hamburg (Germany), and the Paul Scherrer Institute in Switzerland. Each pixel has a size of 200  $\mu\text{m}$  and incorporates a custom-developed ASIC with a dynamic gain switching amplifier [14] and analog memory. The dynamic gain switching allows the detector to provide a high dynamic range with single photon sensitivity at low photon energies (high gain) and about 10,000 photons at energies of 12.4 keV (low gain) [15]. The analog memory is aimed to be capable of storing 352 sensor images during the 600  $\mu\text{s}$  long active train period at 4.5 MHz operation frequency [16]. The system will be used for coherent diffraction imaging and X-ray photon correlation spectroscopy.

**LPD** The *Large Pixel Detector* is developed by a group led by the Rutherford Appleton Laboratory<sup>5</sup> (RAL) near Oxford, provided with contributions from Glasgow University (both UK). The pixel size is in the order of 500  $\mu\text{m}$ . To cover the demand for a high dynamic range, three different gain settings are employed in parallel. Each gain setting has its own analog pipeline and 500 storage

<sup>5</sup>Member of the Science and Technology Facilities Council (STFC)

cells [13], which enable the detector to record up to 1,500 images per bunch train at 4.5 MHz bunch period. The LPD is primarily designed for photons in the 12 keV energy range, but is optionally suited for liquid scattering experiments as well.

**DSSC** The consortium responsible for the development of the *DEPFET Sensor with Signal Compression* detector is led by the Laboratory for Semiconductors (HLL) of the German Max-Planck Institute for Extraterrestrial Physics (MPE) in Munich. Other significant contributors are DESY, Heidelberg University (Germany), Siegen University (Germany), Bergamo University (Italy), and Politecnico di Milano (Italy). Unlike the other detectors with a square pixel geometry, the DSSC is composed of hexagonal pixels with a size of about 230  $\mu\text{m}$ . Custom-developed readout ASICs incorporate an analog filter, an ADC, and SRAM cells for processing data and for local storage of more than 640 images, respectively [17]. The non-linear signal characteristic of the DEPFET sensor allows providing a high dynamic range of up to 10,000 photons at 1 keV per pixel and at the same time single photon resolution at low energies down to 0.5 keV.

## 2.2.4 Data Handling

The experiments at the European XFEL will produce a huge amount of data. Typically, a data rate of about 10 GB/s to 40 GB/s per detector is expected. That is, according to current estimates, operating all six instruments will produce 10 petabytes per year [18]. As a consequence, the development of the DAQ architecture for the European XFEL is mainly driven by the idea to combine several concepts, such as:

### Sufficient data bandwidth

The system must be capable to handle the large data bandwidths produced by the detectors.

### Standard network components and protocols

The networking infrastructure should be based on standard Ethernet network hardware, as commercial components are well-proven and available in large quantities.

### Online data processing

The DAQ system should provide online data processing to reduce the requirements for data storage and data transfer.

### Scalability

Future applications may demand for increasing the number of instruments and larger detector compounds. The system should be capable of being adapted to the increasing number of data links.

### Flexibility

The system should be upgradable when new and faster technologies are available.

As the development of the individual 2d detectors is based on different approaches and technologies, each of them provides a system-specific local DAQ sub-system that most conveniently fits the detector requirements. The data flow of the individual DAQ sub-systems is generalized by a central back-end DAQ system with standardized detector interfaces.

## 3 The DSSC Detector

The DEPFET Sensor with Signal Compression (DSSC) detector is a large format X-ray imager with the capability of megapixel frame readout designed to meet the demands of the European XFEL and its pulse timing structure. The detector is based on DEPFET pixel sensors, which combine photon detector, amplifiers, and electronics to convert the analog signals into a digital representation. The digital values are buffered in in-pixel memory cells, which provide storage capacity for several hundred data values.

This chapter introduces the concept of the DSSC detector as well as its system components and control and readout architecture to provide an overview of the scope of this thesis. The basic characteristics of the DEPFET principle are described in some detail.

### 3.1 The DEPFET Sensor

In 1985, the Depleted P-channel Field Effect Transistor (DEPFET) principle was proposed by Kemmer and Lutz [19], which described a number of interesting characteristics such as built-in amplification, signal charge storage capability, and the possibility of repeated, non-destructive read-out. A few years later in 1990, these characteristics were experimentally verified and confirmed [20]. Since then, a variety of DEPFETs with adapted properties that match the requirements of different applications has been invented. The following description of the DEPFET principle is based on the considerations given in [21, 22, 23].

#### 3.1.1 Functional Principle

The DEPFET principle is based on a p-MOSFET<sup>1</sup> located on top of a high-resistivity, low-doped (ideally fully depleted) n-type silicon bulk, which represents the sensitive volume of the sensor. The bottom of the bulk is covered by a large-area diode serving as radiation entrance window. By applying a sufficiently high negative voltage to the backside p<sup>+</sup> contact, the bulk becomes fully depleted.

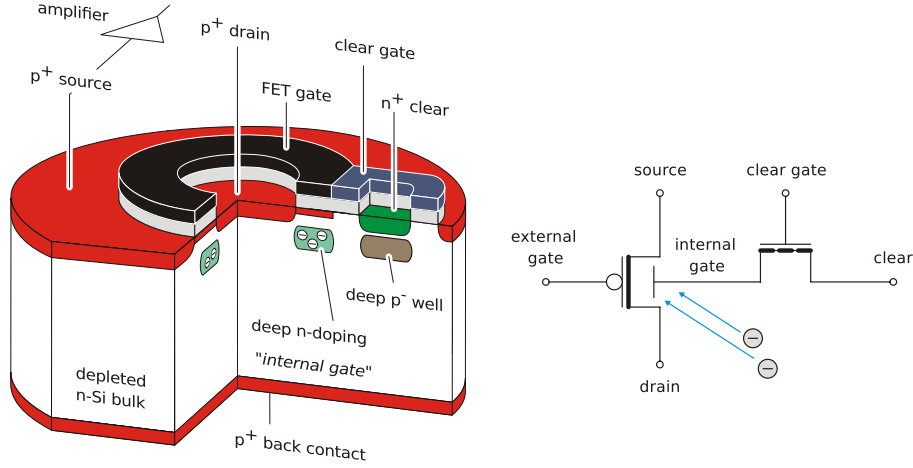
Figure 3.1 shows a schematic view of a DEPFET. A potential minimum for electrons, which is called the *internal gate*, is created below the channel of the transistor by suitable deep n-doping. Electrons, which have been created thermally or by absorption of ionizing radiation anywhere in the depleted bulk, drift to the internal gate and modulate the channel conductivity due to their induced mirror charges. As a consequence, the transistor current is increased for fixed source and external gate voltage, as long as the signal charge is not removed from the internal gate. An integrated n-channel clear transistor<sup>2</sup> enables the electrons to be completely removed from the internal gate by applying a positive voltage pulse to both the clear and the clear gate contact. The difference of

---

<sup>1</sup>Can actually be of MOS type as well as of junction type.

<sup>2</sup>In the original DEPFET proposal, there is only a clear (bulk) contact.

the DEPFET current before and after charge collection, or before and after clearing the internal gate, yields the information about the amount of stored charges, and thus the absorbed energy.



**Figure 3.1:** Three-dimensional view of a DEPFET and its equivalent schematic. Signal electrons are stored in the internal gate, which is a local potential minimum. The induced mirror charges modulate the conductivity of the transistor channel. A clear structure enables the signal charges to be fully removed. Source: [22]

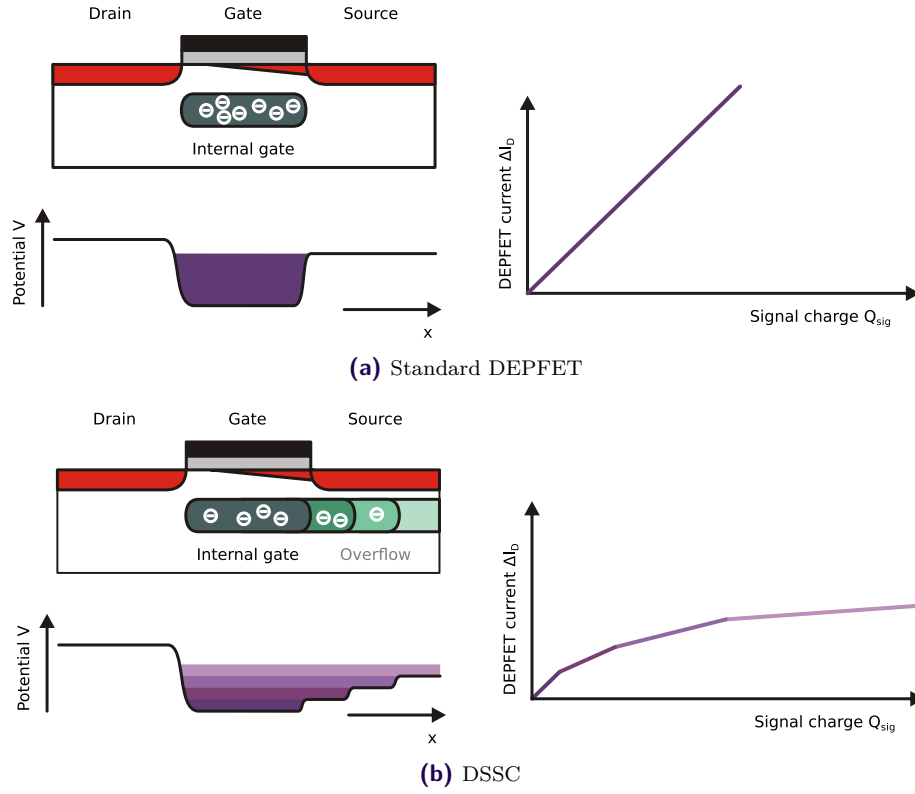
### 3.1.2 DEPFET with Intrinsic Signal Compression

The specifications for the sensor systems of the European XFEL demand for high sensitivity at low signals, as required for single photon resolution, and at the same time providing a high dynamic range to handle several thousand photons per pixel. The DSSC detector employs an enhanced version of the DEPFET sensor, which features an intrinsic signal compression mechanism for large signals. This is accomplished by extending the internal gate beyond the transistor channel into the source region.

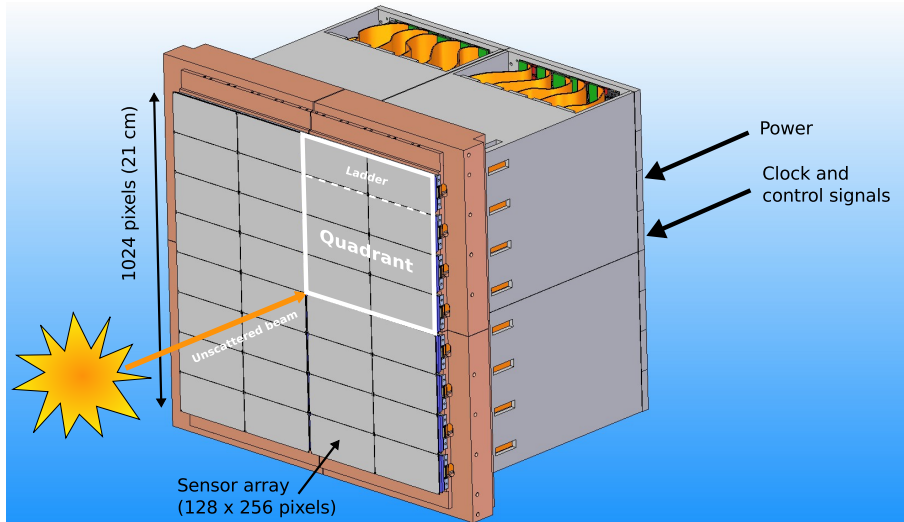
For small signals, the generated charges will cumulate below the channel having a large effect on steering the transistor current. However, large signal charges are only partially stored under the channel, but will also spill over into the extended internal gate in the source region, thus being less effective on the current. As a result, the current-charge characteristic is strongly non-linear for high signal charges, while maintaining linearity for small amount of charges. Figure 3.2 illustrates the cross section as well as the potential distribution and current-charge characteristic of both a standard DEPFET and a DSSC-type DEPFET.

## 3.2 The DSSC Detector

Figure 3.3 shows a three-dimensional view of the DSSC detector. The focal plane is composed of  $1024 \times 1024$  hexagonal pixels of a size of about  $230 \mu\text{m}$  in diameter. The sensitive area is of  $21 \text{ cm} \times 21 \text{ cm}$  in size, subdivided into 16 ladders. Each ladder comprises two monolithic DEPFET sensor arrays with  $128 \times 256$  pixels per array. A sensor array is built out of eight DEPFET sensors



**Figure 3.2:** Schematic view of the cross section and potential distribution of (a) a standard DEPFET with linear amplification and (b) a DSSC with non-linear characteristics. On the right, a qualitative illustration of the current-charge characteristics of the two devices is shown. Source: [22]



**Figure 3.3:** Three-dimensional view of the DSSC detector head. The central hole for the primary beam is not shown here. Source: [17]

with a size of  $64 \times 64$  pixels, arranged two by four. The detector backplane provides the interfaces to external power supplies as well as to the front-end DAQ. The ladders are geometrically arranged such that a central hole is left in the detector plane for the unscattered, high-energetic XFEL beam to go through and get dumped.

The DSSC is designed for photon energies between 0.5 keV and 20 keV. For photons of an energy  $E_{ph} \geq 1$  keV, it provides a dynamic range of more than 10,000 photons per pulse per pixel, while it is still 4,000 for  $E_{ph} = 0.5$  keV. Single photon resolution is supported for photon energies of 0.5 keV. The detector is able to cope with the XFEL timing structure and is optimized for operation at 4.5 MHz electron pulse frequency. In addition, pixel-internal SRAM cells provide storage capacity for more than 640 frames per macro-bunch (train). Table 3.1 summarizes the key characteristics of the DSSC detector.

General	
Energy range	$0.5 \text{ keV} \leq E \leq 20 \text{ keV}$
Dynamic range	$> 10,000$ photons for $E \geq 1 \text{ keV}$ $> 4,000$ photons for $E = 0.5 \text{ keV}$
Resolution	Single photon resolution even at 0.5 keV
Frame rate	0.9 MHz to 4.5 MHz
Frame storage capacity	$\geq 640$ per train
Mean power consumption	$\approx 400 \text{ W}$ in vacuum
Operating temperature	$-20^\circ \text{C}$ optimum, room temp. possible
Focal Plane	
Sensitive area	$21 \text{ cm} \times 21 \text{ cm}$
Number of pixels	$1024 \times 1024$
Pixels shape	Hexagonal
Pixel pitch	$\approx 204 \mu\text{m} \times 236 \mu\text{m}$

**Table 3.1:** Summary of the DSSC detector key characteristics. Source: [17]

### 3.2.1 Detector Concept

The concept of the DSSC detector system is illustrated in the simplified block diagram of figure 3.4. The DEPFET sensors are bump-bonded to mixed-signal readout ASICs (flip-chip mounting), which provide full parallel readout of the pixels. The ASICs are wire-bonded to a Main Board. During the readout phase (inter-train gap), the data stored in the ASIC memory cells are transmitted from the focal plane to the front-end DAQ. Optical high-speed links connect the detector DAQ to the back-end DAQ of the European XFEL, which also provides timing information for and slow control of the detector. A special characteristic of the DSSC detector is the shutdown of most of the analog electronics of both sensors and ASICs during the readout phase for minimizing power consumption. It is accomplished by programmable regulator boards and power switching circuitries.

## Mechanical Layout and System Components

The mechanical layout of the DSSC detector is illustrated in figure 3.5. The camera head of the





**Power Regulator Board** A Power Regulator Board (PRB) provides the power supply regulation for the readout ASICs. In order to minimize power consumption, most of the voltages of both sensors and ASICs are switched off during the 99.4 ms long inter-train gap of the X-ray macro-bunches. Each Main Board is supplied with four PRBs.

**I/O Board** An I/O Board (IOB) is the basic element of the lower front-end DAQ layer. It performs low-level controlling tasks on the detector front-end electronics and provides data from all ASICs of a sensor module to the next DAQ layer. A single IOB manages the control of one sensor module.

**Module Interconnection Board** A Module Interconnection Board (MIB) enables the electrical interconnection between the electronics of the individual system boards of a module.

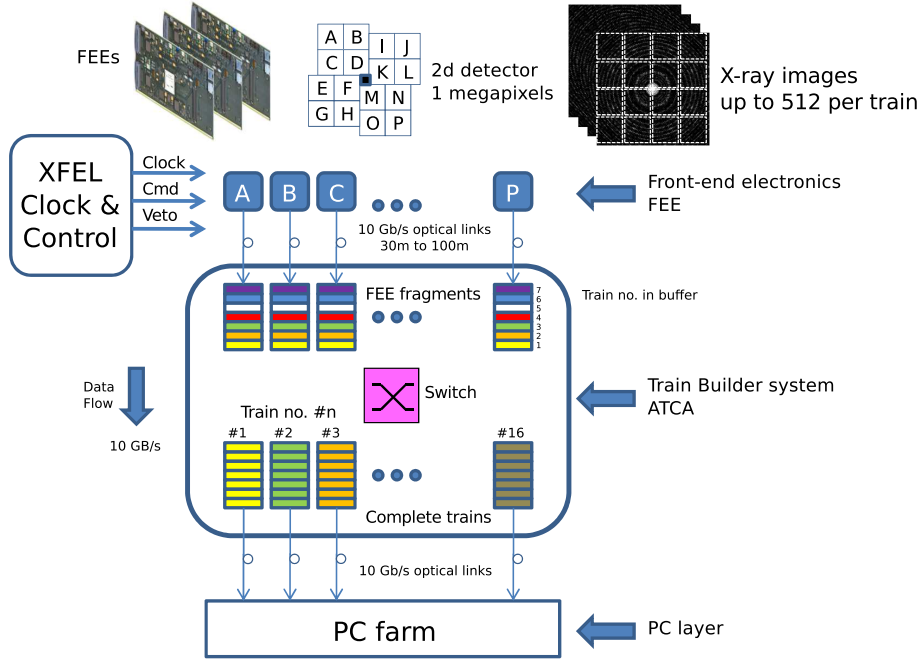
**Patch Panel Transceiver** A Patch Panel Transceiver (PPT) is the basic element of the upper front-end DAQ layer. It acts as a master device to the IOB and provides the interfaces for data, timing and slow-control to the XFEL back-end DAQ. The PPT receives the XFEL timing and generates detector-specific command telegrams that control the DSSC operation. During the inter-train gap, it transmits the recorded image data in return. PPTs are provided on a per-quadrant level.

## 3.3 Control and Readout Architecture

The control and readout architecture of the DSSC system is designed to cope with the the high data rates produced by the detector as well as to comply with the requirements of the demanding bunch timing structure of the European XFEL. The control interface receives the timing information from a copy of an XFEL-general clock and timing distribution system. The detector data, on the other hand, are provided to a back-end DAQ system which is common to all 2d detectors applied at the different experiment stations.

### 3.3.1 Back-end DAQ

The concept of the back-end DAQ of the European XFEL facility is illustrated in figure 3.6. The schematic shows the configuration for a single 2d detector as a representative. The central element is the Train Builder (TB), which is common to all 2d detectors and receives the detector data. The acquired image fragments are merged and transmitted in a pixel-ordered standard XFEL data format to the offline storage facility (PC farm) [24]. Both the detector front-end DAQs and the PC farm interface with the TB over multiple optical 10 Gb/s Extended SFP (SFP+) links. The timing information of the European XFEL trains is distributed to the detectors by the Clock And Control (C&C) system [25]. It provides fast bunch-synchronous commands from the accelerator and implements a vetoing mechanism, which allows discarding unwanted X-ray pulses. As the distribution of the timing information is critical, each of the detectors is supplied with its individual copy of the C&C system.



**Figure 3.6:** Back-end DAQ of the European XFEL. The central Train Builder system acquires data from all 2d detectors and transfers them to the back-end storage facility (PC farm). Timing information of the XFEL bunches are provided through a special Clock and Control system. Source: [24]

## Train Builder

The Front End Electronics (FEE) of the 2d detectors applied at European XFEL facility are expected to have storage capacities of at least 512 images per train, corresponding to an approximate data volume of 1 GB. Considering the readout period of 99.4 ms during the inter-train gap, the total average data rate is in the order of 10 GB/s for a single detector. Data from the detector FEEs are sent to the TB over optical SFP+ links using UDP/IP. The TB is a modular and scalable FPGA-based system, which is implemented in custom Advanced Telecommunications Computing Architecture (ATCA) form factor boards – in the following also referred to as TB boards.

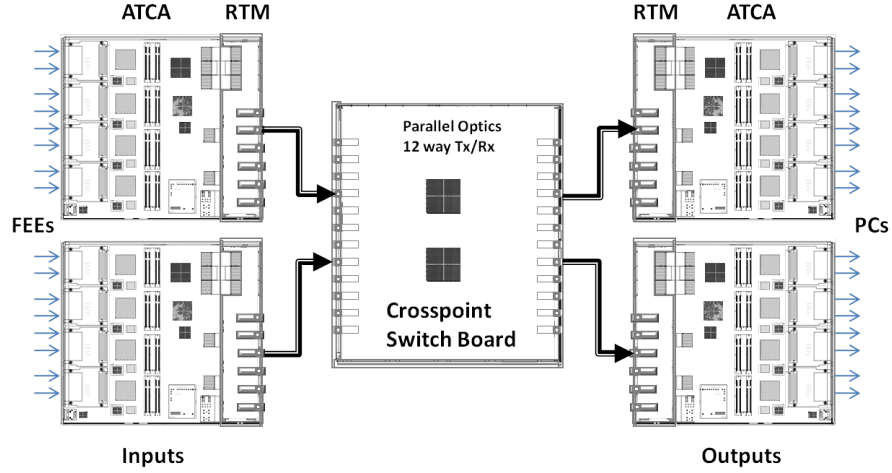
A TB board provides four FPGA Mezzanine Card (FMC) interface cards<sup>3</sup> with two SFP+ transceivers each [26], which represent the optical 10 Gb/s data links to the detector front-end DAQ and the back-end storage, respectively. Each 10G FMC connects to one of the four main I/O FPGAs (Xilinx Virtex-5 FX100T) of the TB board through eight bi-directional high-speed transceiver lanes<sup>4</sup>. The I/O FPGAs connect to four dual DDR2 SDRAM banks supporting up to 16 GB of memory to buffer the incoming detector data. A fifth master FPGA (also Xilinx Virtex-5) synchronizes the four I/O FPGAs and controls a crosspoint switch. The crosspoint switch allows to dynamically connect any of the FPGAs to any neighbor with a switching time of a few microseconds.

The symmetry of the board connectivity allows configuring any of the I/O FPGAs both as a receiver from a detector FEE or as a transmitter towards a computer on the PC farm. While a single TB board provides the readout capability for a quarter megapixel detector, readout systems for larger detectors can be built up using multiple TB boards.

<sup>3</sup>Developed by DESY for the European XFEL

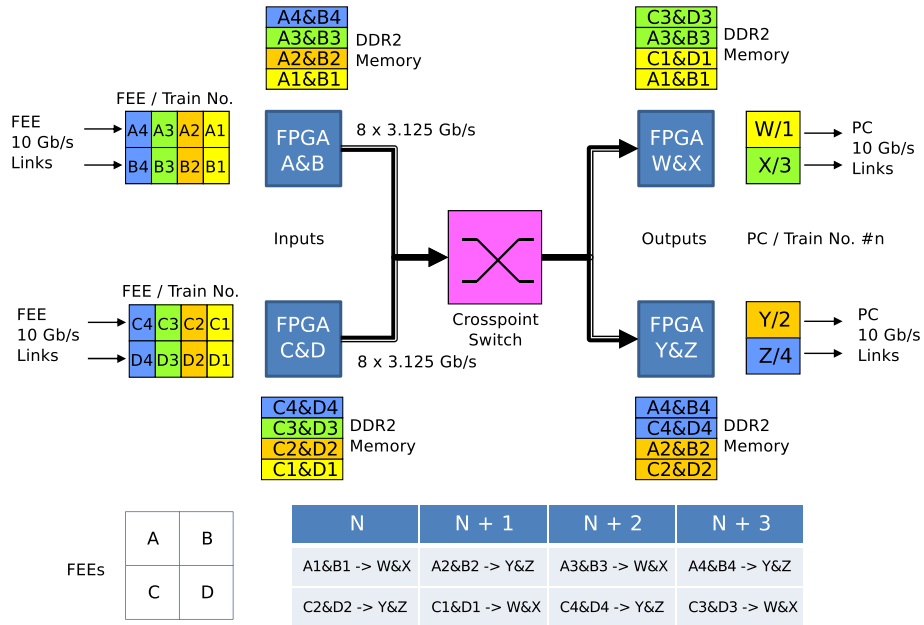
<sup>4</sup>4 x TX and RX per 10G link

The readout system for a megapixel detector can be realized by using four TB boards and an additional crosspoint switch board, as illustrated in figure 3.7. Two ATCA units are configured as receiver only, and two as transmitter only. In figure 3.8, the data flow for the simple case of a



**Figure 3.7:** Realization of a TB readout system for a megapixel detector. The I/O FPGAs of the ATCA boards are configured to act as receiver only (left), and transmitter only (right), respectively. An additional dedicated crosspoint switch board must be supplied to forward the data from the receiving units to the transmitting units. Source: [24]

quarter megapixel detector is shown. The FPGAs A&B and C&D receive data arriving from the

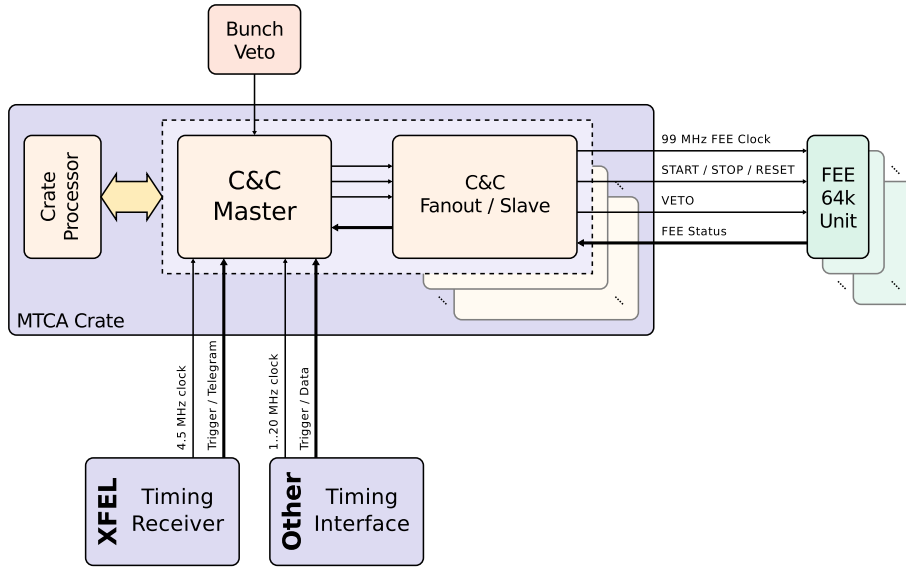


**Figure 3.8:** Illustration of train building with the crosspoint switch for a quarter megapixel system with four optical data links A, B, C, and D. The FPGAs labelled A&B and C&D are configured to act as inputs receiving data from the detector FEEs, while the FPGAs labelled W&X and Y&Z act as transmitters delivering complete images to the PC farm. Source: [24]

detector FEE via two of the 10GbE FMCs. The images arrive in the same time-ordered sequence as the X-ray pulses that generated them and are buffered in the DDR2 memory. At a given cycle  $N$ , the crosspoint switch is configured to transfer data from A&B to W&X, and from C&D to Y&Z, respectively. On the next cycle  $N + 1$ , data from A&B is sent to Y&Z, and from C&D to W&X. That is, a complete detector image is available in the DDR2 buffers of the output FPGAs after two train periods. For an elaborate description of the TB, the reader may refer to [24].

## Clock and Control

The general concept of the Clock And Control (C&C) system is shown in figure 3.9. A Micro



**Figure 3.9:** Schematic representation of the concept of the C&C system. Source: [27]

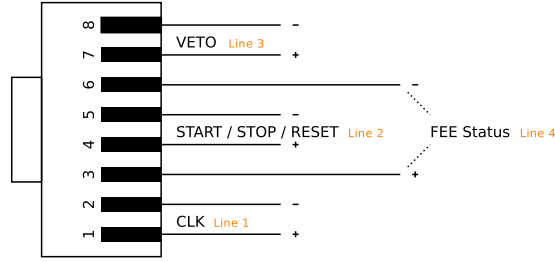
Telecommunications Computing Architecture (MTCA) crate houses a C&C master and several C&C fanout slaves. Both of the main components use the same kind of MTCA boards, which can be configured to act accordingly.

The master takes the 4.5 MHz<sup>5</sup> bunch clock and trigger data as input from the XFEL timing receiver. It additionally receives veto messages provided by bunch veto sources. From these signals, the master generates the 99 MHz<sup>6</sup> FEE clock as well as both the *fast commands* and the *veto telegrams* used to synchronize the detector FEE operations with the global timing of the XFEL facility. The synchronization clock has a fixed phase relation to the bunch clock and is used to *Manchester-code* the fast commands, which are thus delivered source-synchronous.

The number of slave devices is limited by the number of slots of the MTCA crate. The C&C slaves connect to the detector FEEs via RJ45 connectors and standard Ethernet cables. Figure 3.10 illustrates the pinout of the connector. The timing information generated by the master is distributed over three separate lines. Line 1 (CC\_CLK) supplies the 99 MHz *FEE clock*. The fast commands and the veto telegrams are transmitted via line 2 (CC\_CMD) and 3 (CC\_VETO), respectively. A fourth line (CC\_FEESTAT) is optionally used to monitor the status of the detector FEE.

<sup>5</sup>4.5139 MHz

<sup>6</sup>99.3058 MHz =  $22 \times 4.5139$  MHz



**Figure 3.10:** Pinout of the C&C fast signal interface to the detector FEE. The connection is done with an RJ45 connector and ordinary Ethernet network cables. Source: [28]

**Fast Command Protocol** Table 3.2 describes the supported C&C fast commands and their 52-bit wide representation. A *start bit* signalizes the beginning of a transmission and is always '1' for

Command	Start bit	Command bits	Payload	Description
START	1	100	<Train ID> (32) + <Bunch Pattern Index> (8) + <Checksum> (8)	Notifies FEE of incoming train
STOP	1	010	—	Notifies FEE that train ended
RESET	1	001	—	Reset FEE
Reserved	1	111	—	—

**Table 3.2:** Summary of the command telegrams provided to the detector FEEs by the C&C system. Source: [29]

all commands. Three subsequent *command bits* define the command, which can be one out of **START**, **STOP**, or **RESET**. For the start command, a 48-bit *payload* is subsequently transmitted which contains the *train ID* (32 bits), the *bunch pattern index* (8 bits), and a checksum (8 bits). The start telegram is delivered 15 ms prior to the first arriving bunch. In that way, there is enough time to initialize the detector electronics and prepare the systems for capturing the data.

**Veto System** As the sensors produce pixel data for every single incoming bunch, the detector FEEs have to deal with a huge data volume during the macro-bunch period. On the other hand, present detector technology limits the number of possible data sets being transferred from the detector front-end to the back-end DAQ to only a fraction of the 2,700 pulses. A number of at least 512 pulses, however, is expected to be stored in the FEE ASIC pipelines. Therefore, the detectors are designed to provide a way to remove unwanted data sets and free space for other, possibly better images in order to use the limited storage capacity in a much more efficient way. This mechanism is called *vetoing mechanism*, or simply veto. A bunch can be classified into three veto categories:

#### VETO

A particular bunch is classified as VETO if the measurement of this bunch was most likely not successful. The memory location, which the data of this bunch is stored in, should be freed for reuse.

#### GOLDEN

If a certain bunch is classified as GOLDEN, there is a high possibility that the measurement

for this bunch was very good. The corresponding memory location should be marked as write-protected to ensure the data are definitely kept.

### NOVETO

A particular bunch is classified as NOVETO if it can be assigned to neither VETO nor GOLDEN. It is the default veto class. Different handling of a NOVETO is possible depending on the VETO policy implemented by the individual detectors.

The veto mechanism is implemented as a low-latency protocol. It is provided by the C&C system to the detector FEEs over a transmission line separate from the C&C fast command line. Table 3.3 summarizes the bunch veto classifications and their telegram representation in the veto protocol. A

Command	Start bit	Command bits	Payload	Description
VETO	1	10	<Bunch ID> (12) + 0000	Identifies bunch ID to be vetoed
NOVETO	1	01	<Bunch ID> (12) + 0000	No veto defined for given bunch ID
GOLDEN	1	11	<Bunch ID> (12) + 0000	Identifies given bunch ID as "golden"
Reserved	1	00	<Bunch ID> (12) + 0000	–

**Table 3.3:** Summary of the veto telegrams provided to the detector FEEs by the C&C system.  
Source: [29]

veto telegram consists of 19 bits. The very first bit transmitted is a *start bit* and is always '1'. Two *command bits* define the veto class, followed by the 12-bit wide *bunch ID*. The last four bits are reserved for future use and are set to "0000".

The veto protocol is available in two slightly different schemes, since the 2d detectors have different implementations of veto policy. The *fixed delay protocol* was primarily designed to fulfill the requirements of the LPD system. The veto telegram for a specific bunch arrives with a fixed latency to the bunch detection. Its main characteristics are:

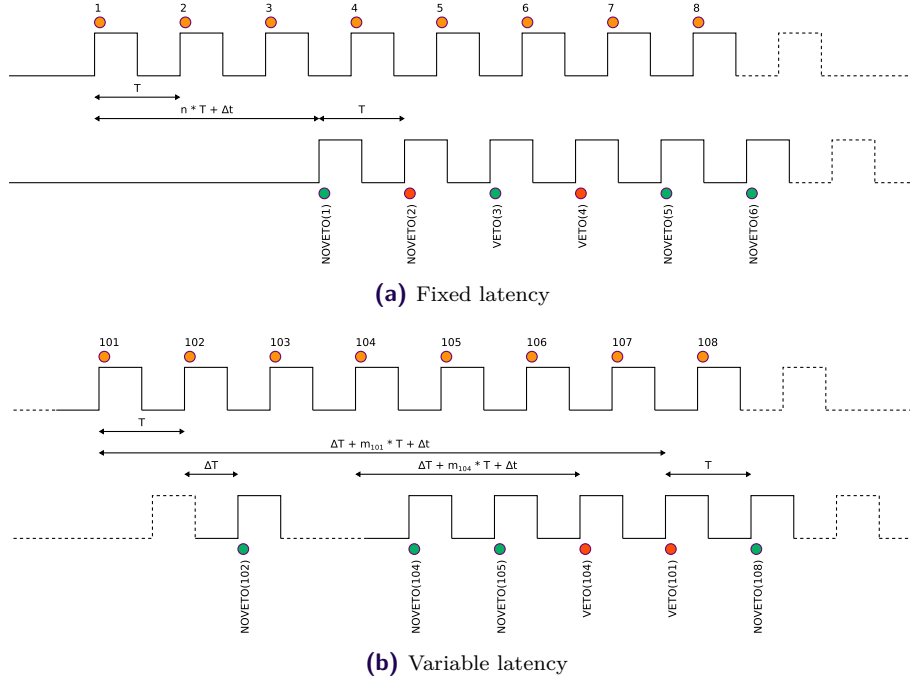
- The delay of a veto telegram is the sum of
  - a system-dependent delay  $\Delta t$  and
  - a configurable, but constant delay  $n \cdot T$  ( $n = 0, 1, \dots$ ), which is specifically configured for a detector
- The order of veto telegrams is monotonic and increases with the bunch number (that is from #1, #2, ..., #n).
- The number of telegrams transmitted matches the number of bunches in the train.
- Once a veto telegram has defined the class of a particular bunch, it is fixed and unchangeable.

The *variable delay protocol* will be used by AGIPD and DSSC. In contrast to the fixed delay protocol, it allows independent delays within a certain, detector-specific range. As a consequence, the storage cells of a detector can be used much more efficiently. Further key characteristics are:

- The delay of a veto telegram is the sum of
  - a system-dependent delay  $\Delta T + \Delta t$  and
  - a variable delay  $m_i \cdot T$ , where  $m_i = 0, 1, \dots, m_{max}$  denotes variable parameter specific to the bunch number  $i$

- The bunch IDs of NOVETOs are always increasing and reflect the actual state of the bunch counter.
- The order of VETO and GOLDEN telegrams can be non-monotonic (i.e. #104 before #101).
- Information for a given bunch ID can be received twice (VETO or GOLDEN after NOVETO).

An example of the protocol for both fixed and variable latency is given in figure 3.11.



**Figure 3.11:** Examples of the veto protocol for (a) fixed and (b) variable latency implementation. Source: [28]

### 3.3.2 DSSC Front-end DAQ

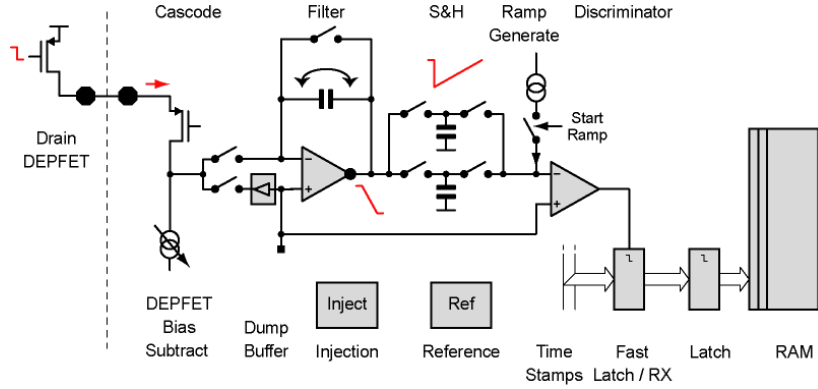
The DSSC front-end DAQ comprises the data chain from the output interface of the readout ASICs to a standardized input interface of the central back-end DAQ system of the European XFEL, which is based on optical 10 Gb/s links. The readout sub-system also includes the signal chain required to provide timing and control information from the XFEL system to the detector. Its functionality is implemented on two FPGA-based components – the Patch Panel Transceiver and the I/O Board. The concept and implementation of the front-end DAQ system for the DSSC detector is elaborately described in the next chapter.

### 3.3.3 Readout ASICs

The DSSC readout ASIC is realized as a 4k-pixel chip with  $64 \times 64$  channels bump-bonded to a DEPFET sensor. The entire one-megapixel DSSC detector comprises 256 of those readout devices. The ASICs are designed for low-noise readout of the DEPFET current and at 4.5 MHz operation



frequency [30, 17]. Figure 3.12 shows a simplified block diagram of one DSSC pixel chip. The



**Figure 3.12:** Simplified block diagram of the readout ASIC for the DSSC detector. Source: [30]

signal current from the sensor is injected into a *cascode* to keep the DEPFET at constant potential. This reduces both the speed effect of the drain capacitance and the gain effect of the low output resistance of the sensor. A *bias subtract circuit* compensates for the DEPFET bias, which is fixed by the gate-source voltage, but can vary from pixel to pixel. The *filter stage* (Politecnico di Milano) integrates the DEPFET drain current to a voltage before and after the arrival of a single bunch and outputs the difference between the two consecutive integrations [31]. By flipping the feedback capacitance, it is possible to accomplish both the integration phase (baseline readout) and the de-integration phase after the flat-top (baseline plus signal readout) by a single stage only. The resulting weighting function is of trapezoidal shape. The digitalization of the filter output is realized by a *Wilkinson-type single-slope ADC* with 8-bit resolution<sup>7</sup> developed by DESY in Hamburg [32]. It translates the output voltage of the filter stage into a timing information. A pair of *sample and hold capacitors* is used to pipeline the signal processing by alternately connecting them in a cross-wise manner. While one capacitor is charged by the output of the filter for a bunch  $n + 1$ , the other one that holds the signal voltage for bunch  $n$  is discharged by a programmable constant current source. A precise strobe starts the generation of the voltage ramp and also triggers a Gray-code counter to run. The counter toggles on both rising and falling edges of a fast 695 MHz clock, yielding a time resolution of about 719 ps. Once a reference potential is reached, a comparator latches the 8-bit Gray-code counter time stamp. The integrated *SRAM cells* provide storage capacity for at least 640 ADC time stamps of 9 bits during an XFEL train. The memory access is implemented such that individual memory locations can be overwritten, which is the base for the vetoing mechanism. In addition, the ASIC contains several auxiliary blocks, such as static control registers, a large and switchable capacitor, and monitoring lines. An *injection circuit* (developed by Universita di Bergamo) allows supplying a known programmable current into the input node for calibration and health check purposes [33]. A global logic, which implements a Finite State Machine (FSM), controls both the data taking sequence (during active train period) and the serial readout of the SRAM cells (during inter-train gap).

<sup>7</sup>9-bit for operation at 2.2 MHz



## 4 FPGA-based DAQ Systems in Modern Physics Experiments

The detectors applied in modern physics experiments usually produce great quantities of data, which are distributed over a multitude of readout channels (several thousands up to several millions). Accumulating the produced data from the channels and delivering them to a storage facility for further offline analysis is the main purpose of a Data Acquisition (DAQ) system. Apart from handling the large amount of data and input channels, however, the readout system has to face several other challenges, such as high event rates and performance requirements for event processing. Moreover, there are also demands for high connectivity and throughput on the network infrastructure. Considering other aspects like scalability, flexibility, and the financial aspect have led to a growing application of Field Programmable Gate Array (FPGA)s in DAQ systems.

The major advantage of FPGA-based systems is the fast and massive-parallel processing of data using a dense implementation while yet preserving flexibility. The reconfigurable logic of an FPGA allows adapting to application-specific changes of the interfacing system (within certain boundaries) during development or even at runtime without the need to change the underlying hardware layout.

This chapter introduces into FPGA-based data acquisition systems and the benefits of their application at detector systems of modern physics experiments using the example of the Large Hadron Collider (LHC) at CERN. The second part describes the concept and hardware realization of the front-end DAQs of both AGIPD and LPD systems at the European XFEL in more detail.

### 4.1 DAQ Systems at CERN

At the LHC at CERN, the European Organization for Nuclear Research, all of the four large high-energy physics experiments ALICE, ATLAS, CMS, and LHCb use custom FPGA-based components for implementing trigger and DAQ systems<sup>1</sup>. The requirements are different with regard to the trigger rate, the event size, and the input and output data rate. The following overview is founded on the descriptions given in [34] for ATLAS, CMS, LHCb, and ALICE, [35] for LHCb, and [36] for ALICE.

The trigger system for CMS is implemented in two stages. The first level trigger L1 combines the data of all detector channels (approx. 55 millions) into 626 event fragments, which reduces LHC bunch-crossing rate of 40 MHz to a 100 kHz L1 event rate. As the event size is about 1 MB, the total data volume being transferred to the second trigger level (HLT) is in the order of 100 GB/s. The HLT algorithms further filter the events and send them to the storage facility at a rate of 100 Hz. In order to obtain the required bandwidth, the CMS uses 458 custom FPGA-based readout modules.

---

<sup>1</sup>The experiments at CERN make use of a multi-staged trigger system. The so-called High Level Trigger (HLT) is common to all systems and represents the final trigger stage.

The requirements on the DAQ of ATLAS are very similar to those of CMS. Its first level trigger L1 also reduces the LHC bunch rate to an event rate of 100 kHz. Data from approx. 100 million detector channels are concentrated into 1,600 readout modules. The typical event size is about 1 MB, yielding a total data volume 100 GB/s. As opposed to CMS, ATLAS introduces an intermediate trigger level L2, which further decreases the event rate to 1 kHz, before the HLT mechanism performs event selection for storage at a rate of 100 Hz. ATLAS employs several hundred custom modules based on FPGA technology to meet the requirements on bandwidth and event processing.

The characteristics of LHCb and ALICE are different with regard to event sizes and bandwidth requirements. LHCb completely omits a first level trigger stage for lowering the event rate<sup>2</sup>. Instead, data of about 276,000 detector channels are fed into 90 FPGA-based front-end readout units with 3,072 input links each. The size of an event is approx. 35 kB at a rate of 1.1 MHz maximum, which leads to a total front-end input data volume of about 337 GB/s. The overall throughput to the DAQ network amounts to 6.4 GB/s at 2 kHz rate after processing the data in the readout units.

ALICE uses a four-staged trigger system. The first level trigger L0 has to cope with relatively low input event rates of 5 kHz, however, as the event sizes can be as large as to 85 MB, the DAQ system must be designed for a maximum input bandwidth of 425 GB/s. The event rate is gradually reduced by the subsequent trigger stages to 400 Hz (L1), 200 Hz (L2), and finally 100 Hz (HLT). Considering the event size, the maximum expected output data rate is in the order of 8.5 GB/s.

## 4.2 DAQ Systems for the 2d Detectors at the European XFEL

The front-end DAQ systems applied at the European XFEL must be capable to cope with the demanding electron bunch timing of the facility, and also with the large data volume produced by the detectors. The back-end DAQ, in addition, expects pixel-ordered image data, which demands for online pre-processing (re-ordering) of the recorded pixel data. The readout sub-systems generally have to provide the following functionality:

- Accept the timing information from the C&C system and generate appropriate signals for the detector system
- Collect and pre-process the sensor data sampled by the readout ASICs, and provide them to the corresponding copy of the TB system
- Provide the mechanism required for remote slow-control of the detector

For realization of a system that reliably fulfills these requirements, several fundamentally different options with respect to the underlying technology exist.

### Standard PC technology

A system exclusively built from commercial standard PC components would require to connect the system with the detector and provide interfaces for both timing / control and data readout. Standard networking cards could be a solution. However, the latency induced by those components makes such a system inapplicable for handling the timing restrictions. In addition, standard components could most likely not be interfaced with the application-specific detector electronics.

---

<sup>2</sup>The first trigger level stage of LHCb is called L0.

**PCs extended by a custom peripheral board**

A custom peripheral board (e.g. a PCIe card) could be a feasible implementation concept for connecting a standard PC with the detector FEE. The timing could be managed by this extension board with a minimum latency. Modern 10GbE network cards provide sufficient bandwidth to interface with both the readout chips and the back-end DAQ. However, since modern CPUs lack of massive parallelism, they are not well-suited for pre-processing the huge amount of data. As an option, the processing could be outsourced to GPUs which provide parallelism. However, the additional system complexity and reliability issues of standard PCs would not justify the effort and costs to build and maintain such a system.

**Custom PCB with dedicated ASICs**

The main reason a system setup with dedicated ASICs is not feasible lies in the fixed implementation of the processing algorithm. Since the objectives of an experiment and its processing scenarios could evolve, it is desirable to make use of a solution which provides a certain amount of flexibility. Also, the development and production of ASICs usually involve high costs.

**Custom PCB with FPGAs**

An FPGA-based system offers several advantages over the previously listed implementations. FPGAs provide massive parallel processing of data, which makes them very efficient devices. Due to their reconfigurable logic, a high amount of flexibility is also given. Furthermore, modern FPGAs are much less power-consuming (compared to GPUs, for example) and allow a dense implementation. Lastly, the production costs for FPGA-based systems are within a reasonable range.

Given these considerations, an FPGA-based system is the best solution for DAQ systems at the European XFEL.

**4.2.1 The DAQ at AGIPD**

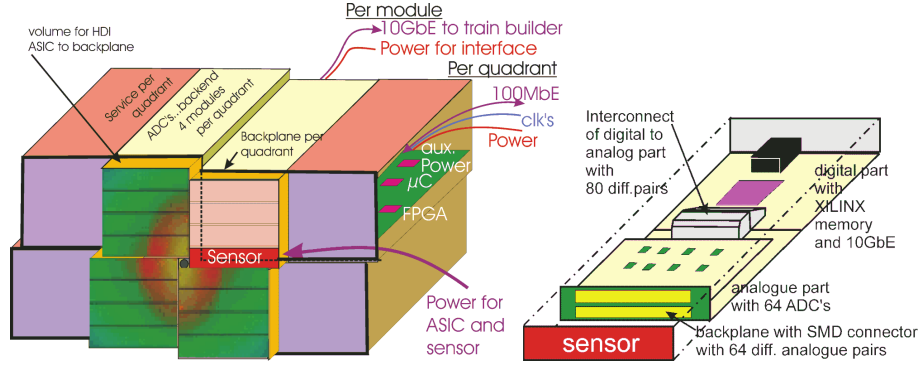
The following paragraph explains the front-end DAQ of the AGIPD system based on the descriptions given in [37, 38].

The focal plane of the AGIPD device is divided into 16 sensor modules grouped into four quadrants. A sensor module houses 16 readout ASICs, which are bump-bonded to the sensors and connected to a backplane via high-density interfaces. The ASICs provide storage capacitors to buffer the analog pixel data of 352 bunches during a train. Behind the backplane resides the interface electronics of a module. It provides the required signals to control the ASICs, digitizes the analog pixel data stored in the buffers, and generates the data streams that are transmitted to the back-end DAQ system. The functionality for controlling both the ASICs and the readout and processing of the pixel data of the interface electronics is distributed across two kinds of modules, the control module and the readout module.

The control module interfaces to both the C&C system and the slow-control and is located next to the focal plane of each quadrant. An FPGA receives the C&C timing information and generates detector-specific control signals, which are distributed on a per-quadrant level. The slow-control functionality is implemented using an ARM microcontroller, which runs a Linux operating system for embedded systems. The microcontroller communicates with the control system over 10/100 Mb TCP/IP Ethernet.

A readout module is responsible for processing the data of one sensor module and resides directly behind the backplane. The readout module comprises two sub-boards for handling the analog

pixel data with filters and ADCs, and another sub-board for processing the digital data using a Xilinx Virtex-5 FPGA. The digital part also interfaces with the TB via one optical 10 Gb/s SFP+ transceiver. Figure 4.1 illustrates the concept of the detector system and its DAQ electronics. The



**Figure 4.1:** Illustration of the detector concept of AGIPD (left) and its front-end DAQ electronics (right). Source: [37]

analog pixel data stored in the buffer capacitors of the 16 ASICs are digitized during the readout phase by 64 ADCs of type AD9252 (Analog Devices), an octal 14-bit and 50 MS/s device<sup>3</sup>. The digitized data are captured by the readout FPGA via 64 electrical links, yielding a total data rate of  $64 \times 700 \text{ Mb/s} = 45 \text{ Gb/s}$  that must be handled by the FPGA. Data reduction is applied to narrow the total data rate down to 5 Gb/s per module. The data are sorted to the standard order and buffered in DDR2 memory, before they are transmitted towards the TB over the optical 10 Gb/s SFP+ interface. Table 4.1 summarizes the signal and data bandwidths inside the AGIPD detector head.

	Devices / links per			Data rate per	
	module	detector head	link	module	detector head
ASIC	16	256	Pixel data generation		
ADC	64	1024	50 Mb/s	3.2 Gb/s	51.2 GS/s
LVDS TX	64	1024	700 Mb/s	45 Gb/s	720 Gb/s
FPGA	1	16	Data reduction		
SFP+	1	16	5 Gb/s	5 Gb/s	80 Gb/s

**Table 4.1:** Summary of the payload data rates inside the AGIPD front-end DAQ. The massive data volume of 45 Gb/s per module is narrowed down by applying data reduction algorithms inside the FPGA. Source: [37]

## 4.2.2 The DAQ at the LPD

The principle of the LPD front-end DAQ is shown in figure 4.2. The detector is built out of 16 identical so-called supermodules, each carrying 128 readout ASICs. A Front-End Module (FEM), which is an FPGA-based DAQ card developed by STFC Rutherford, represents the key component of the LPD front-end DAQ and is mounted on the rear of each supermodule. It implements both control and readout functionality for the ASICs and provides the interfaces to the TB and the C&C

<sup>3</sup>The term MS/s refers to "mega-samples per second"



proper pulse arrival time order<sup>5</sup> by programming the DMA engine accordingly. The sorted image data are sent towards the TB using a VHDL module, which is based on the Xilinx XAUI core and interfaces the Virtex-5 device to the optical transceiver FMC.

Both the clock and the command telegrams from the C&C system are received and processed by the main FPGA. It generates and broadcasts LPD-specific commands to all ASICs of a supermodule, which trigger bunch recording and data readout, or instruct the ASICs which bunches to store based on the veto information received.

---

<sup>5</sup>The data will be disordered due to the operation of the ASIC veto pipeline logic



## 5 The DSSC Front-end DAQ

The functionality of the DSSC readout chain is distributed across two components with the PPT as master component and the IOB as slave module. The digital part of the ASICs, which implements the sampling, buffering, and also the transmission of the pixel data, can be considered as a third DAQ level, although the ASICs are part of the sensor front-end electronics.

The first part of this chapter elaborately presents initial design considerations and the concluded concept of the DSSC front-end DAQ. Subsequently, the hardware implementation of the basic DAQ modules is described, before the chapter is completed with a comparison of the three DAQ systems applied at the European XFEL.

### 5.1 The DAQ at the DSSC

The DSSC DAQ electronics comprises the data chain from the output interface of the readout ASICs to a standardized optical input interface of the XFEL back-end DAQ. Also, the DAQ sub-system includes the signal chain required to provide timing and control information from the XFEL system to the detector.

#### 5.1.1 The Concept

During the design process, a number of different concept options were reviewed that are described briefly, before the conclusive decision is presented. All three implementation options presented in the following consider a solution with FPGAs based on the motivation given in the previous chapter.

#### Baseline Implementation

The initial proposal for the DSSC DAQ foresees a uniform DAQ box based on high-end PCs that incorporate a custom peripheral board with FPGA co-processors. The PC allows creating a uniform interface to the XFEL back-end system, while the reconfigurable hardware allows adapting the DAQ box to various detector requirements. Data transmission towards the TB is realized using optical 10GbE links provided by a customizable I/O extension board for the FPGA co-processor. For timing and control, another FPGA co-processor board of the same type interfaces with the C&C system.

The system was intended for being realized with the MPRACE-2 FPGA co-processor board developed at Zentrales Institut für Technische Informatik (Central Institute for Computer Engineering) (ZITI) of Heidelberg University [40]. Custom I/O mezzanines would provide the electrical and optical interfaces to the detector modules and the back-end DAQ, respectively.

The proposal was rejected since the concept of the detector FEE evolved to a degree at which the

DAQ could not conveniently be realized with this approach. The concerns of latency and stability issues using standard PC technology made another point in the decision.

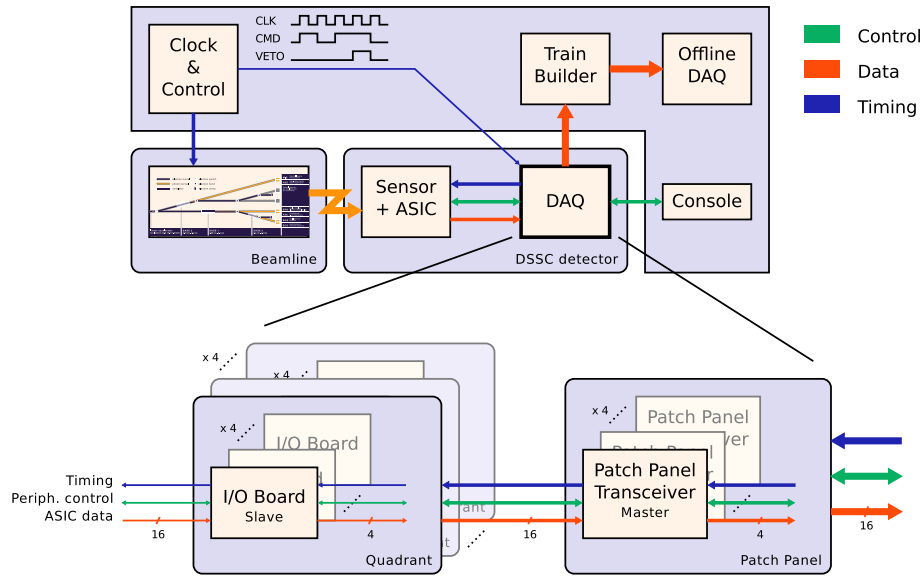
## Local Transceiver Implementation

The second draft of the front-end DAQ concept proposes an implementation solely based on local FPGA transceiver elements. Each transceiver unit directly connects with one detector module and handles its operation, that is both control / timing and data acquisition. Interfacing with the back-end DAQ is realized by one optical 10GbE link and one electrical link for transmitting the data towards the TB and receiving the timing information from C&C, respectively.

Although this concept deals with the latency and stability issues as well as the practical flaws of the initial proposal, the evolving specifications of the detector FEE required adapting this approach, which finally led to the following implementation.

## Extended Transceiver Implementation

Figure 5.1 illustrates the final concept of the front-end DAQ system for the DSSC. It is an advanced



**Figure 5.1:** Illustration of the concept of the DSSC front-end DAQ. The two main components are the PPTs and the IOBs, which connect to both the readout ASICs and the back-end DAQ. The red arrows designate the data path from the sensors to the offline storage facility. The timing distribution and the remote control are illustrated by the blue and green arrows, respectively.

version of the local transceiver implementation, but uses a two-staged approach instead. In that way, a master-slave concept is introduced, in which the masters provide global detector management, while the slaves operate on a module level.

The DSSC front-end DAQ connects to both internal (sensor and ASIC) and external (C&C, TB, and slow-control operator console) interfaces through the IOBs and the PPTs, respectively.

A PPT as the basic element of the upper DAQ layer supervises one detector quadrant. Each PPT receives its own copy of the C&C control and timing signals. From the 99 MHz FEE clock,

which serves for synchronization with the electron bunch timing, the PPT derives the 695 MHz sampling clock that is provided to the ASIC ADCs. The control and veto telegrams are decoded and converted into detector-specific commands, which are distributed to and processed by the master FSMs of the ASICs. A PPT concentrates the image data of a full quadrant (i.e. 64 readout ASICs) into four optical 10 Gb/s links that interface with the TB system. In addition, the PPT provides the mechanisms for remote slow-control from the operator console.

An IOB as the basic element of the lower DAQ layer is responsible for controlling the peripheral electronics of a single sensor Main Board. Moreover, the IOB FPGA provides the readout logic to capture the pixel data from the 16 readout ASICs of an Main Board and concentrate them into four<sup>1</sup> electrical 3.125 Gb/s high-speed links for transmission towards the PPT. The main controlling purpose is the shut-down of analog (and partially digital) power for the sensors and ASICs for reasons of saving power during the readout phase.

A note on the naming of the FPGA high-speed transceivers. Depending on the FPGA chip family, a transceiver is referred to as Multi-Gigabit Transceiver (MGT), Gigabit Transceiver at low Power (GTP), and Extended Gigabit Transceiver (GTX). As a general term, high-speed Serializer / De-serializer (SerDes) is also used throughout this document.

### 5.1.2 Timing and Control of the DSSC DAQ

The timing and controlling tasks of the DAQ are distributed to its main constituents on a per-quadrant level as follows:

#### Timing

The global DSSC timing is provided by the PPT by decoding the C&C telegrams and translating them into DSSC-specific timing signals. However, each subjacent IOB additionally generates a local timing synchronous to the global timing structure for different purposes like data capturing, sequencing the power switching, and steering critical sensor signals.

#### ASIC Fast-control

The instructions for controlling the ASIC readout are explicitly generated by the PPT. The IOBs receive a copy of these signals to trigger the data capturing process accordingly.

#### Slow-control

Slow-control refers to the mechanisms of configuration, monitoring, and maintenance of the DAQ system. It is implemented on the PPT by using a combination of microcontroller and embedded software that communicate with control registers of both the PPT and its assigned IOBs.

### Timing

The important task of the local DAQ system regarding timing is to synchronize the detector readout electronics with the global electron bunch timing provided by the XFEL machine such that sampling and digitization start with a well-defined, low-jitter phase relation with respect to each bunch. At the level of the PPT, the operation is defined by the following states:

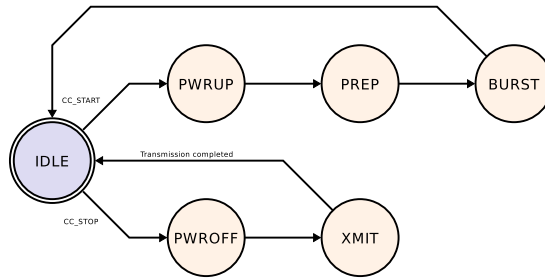
- **IDLE:** Entered after power-up or reset signal by C&C; reset signal is forwarded to IOB and Main Board

---

<sup>1</sup>Recent DAQ development uses only three high-speed links for practical reasons.

- **PWRUP**: Entered from **IDLE** after C&C start command; sends telegrams to IOB to power up sensor and ASICs; proceeds to **PREP**
- **PREP**: Entered from **PWRUP**; inserts a delay for defined number of cycles to let power stabilize; proceeds to **BURST**
- **BURST**: Entered from **PREP**; sends telegram to IOB and Main Board to indicate first incoming bunch after a defined number of cycles; proceeds to **IDLE**
- **PWROFF**: Entered from **IDLE** after C&C stop command; sends telegram to IOB to turn off sensor; proceeds to **XMIT**
- **XMIT**: Entered from **PWROFF**; sends telegram to IOB and Main Board to start data transmission; delay for known transmission time; proceeds to **IDLE**

Figure 5.2 visualizes the possible state transitions. The timing generated inside the DSSC DAQ is



**Figure 5.2:** Preliminary state transition graph of the PPT timing FSM.

distributed to the ASICs over three signals.

The most critical signal of the control path is **ADCCLK**, which is the clock for the Gray-code counter of the ADCs. In order to maintain the precision of the DSSC, the jitter of the clock signal must be as low as possible. It is also required that **ADCCLK** has a known phase relation to the **FEE** clock generated by C&C, and thus to the XFEL bunch clock. Therefore, **ADCCLK** is generated by a low-jitter PLL that can be programmed with a defined phase relation between the input and output clock signals. The PLL multiplies the **FEE** by a factor of seven, yielding a frequency of about 695 MHz<sup>2</sup> for the Gray-code counter. The serial ASIC data output links run at a frequency half of **ADCCLK** (approx. 350 MHz).

Signal **XCLK** is a copy of the C&C **FEE** clock itself and is used for sampling the ASIC command telegrams, which are transmitted over the **XDATA** signal line and are synchronous to **XCLK**.

A fourth control signal **XRESET** enables globally resetting the entire detector quadrant, including the FPGA logic on both the PPT and the IOBs as well as the sensors and the ASICs, respectively.

## VETO Mechanism

The DSSC DAQ will implement a RAM-based vetoing mechanism that supports *variable delay*. Possible incoming veto commands are sorted to chronological order by the PPT, and corresponding telegrams are sent to the IOB and the Main Board with a defined delay with respect to the veto bunch number. However, vetoes do not impose a state transition in the PPT timing FSM. To provide maximum flexibility, a small memory is initialized during the preparation phase containing

---

<sup>2</sup>695.1406 MHz = 7 × 99.3058 MHz

*valid* or *bunch-reject* information for each bunch, according to the actual bunch pattern. The memory is read during the train with an initial delay corresponding to the maximum allowable veto latency. Subsequent veto telegrams modify the memory pattern such that bunches are rejected according to both pre-defined bunch pattern and actual veto commands. Internally, the veto mechanism operates at fixed latency.

## ASIC Fast-control

The DSSC detector features 256 readout ASICs in total, which are controlled quadrant-wise by four PPTs. Each PPT receives a copy of the timing information from the C&C system and locally generates detector-specific control commands for the ASICs. The commands instruct the ASICs to power-up the analog electronics and start data recording, reject a bunch in case of a veto, and start data readout, respectively.

The PPTs provide a copy of the four local timing / control signals (ADCCLK, XCLK, XDATA, and XRESET) to each of the 16 IOBs in parallel. On the IOB, the clock signals are duplicated for the two Main Board sections by fanout buffers, before another stage of fanout buffers residing on the Main Board broadcast them to each individual readout chip. On the other hand, both the data and the reset signal are provided as single lines across the Main Board that connect the ASICs via star topology.

The command telegrams are transmitted serially and captured by the master FSM of the ASICs. In order to ensure that the FSM always properly recognizes an incoming telegram, it is required that XCLK is in phase with ADCCLK. An overview of the commands supported by a readout chip is given in table 5.1. The protocol is implemented as 5-bit commands. The first bit is a start bit and

Command	Start bit	Command bits	Description
START_BURST	1	0000	Indicates start of new burst (train)
START_READOUT	1	0001	Starts data readout
VETO	1	0010	Vetoes an event with fixed latency
START_TESTPATTERN	1	0011	Starts sending the test data pattern
STOP_TESTPATTERN	1	0100	Stops sending the test data pattern

**Table 5.1:** List of the fast control commands supported by the ASIC master FSM. Source: [41]

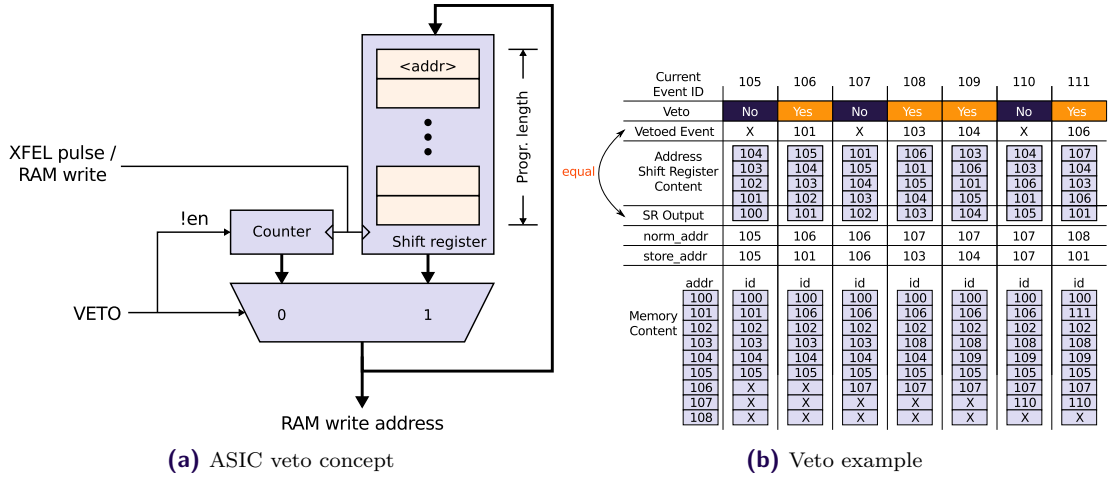
is always '1'. The subsequent four bits define the actual command that triggers a certain state transition of the ASIC FSM. At the beginning of a train period, **START\_BURST** triggers powering the analog electronics and configuration of the ASICs. Subsequently, data are recorded according to the bunch pattern information provided by the C&C system. When the macro-bunch period ends, **START\_READOUT** initiates the data readout. The ASICs serially transmit the content of their memory cells towards the IOBs, which concentrate and forward the pixel data to the PPT for further processing and transmission towards the TB.

Vetoing an event is signaled by the **VETO** command. Inside the ASIC, the veto mechanism is implemented with a fixed latency. An address counter generates incrementing addresses starting at zero. A shift register that acts as a FIFO keeps track of which addresses have already been written to. Its content is shifted by one every bunch cycle. The shift register has a programmable length. The maximum length  $n_{max}$  reflects the maximum allowable veto latency in number of bunch cycles passed. However, the length is fixed for an entire train.

Supposing no vetoes have been issued during the train, the address from the counter is used to

determine the next memory cell to be written. When a **VETO** command is issued for a certain bunch, the counter pauses, and the address pending on the shift register output is used instead. Since every buffered address has to traverse the full shift register, the output of the shift register always points to the memory location that was written to  $n_{max}$  events ago. As a consequence, incoming vetoes will disturb the monotonic address order.

In order to provide a variable latency behavior, the PPT intercepts the vetoes from C&C and bloats the latency to  $n_{max}$  in case a veto has arrived for a bunch earlier than  $n_{max}$  events ago. Vetoes for bunches longer ago than the maximum latency are dropped by the PPT. Figure 5.3 demonstrates (a) the concept of the vetoing mechanism and (b) an example for a veto latency  $n_{max} = 5$ . Up



**Figure 5.3:** Concept and example of the DSSC veto mechanism as implemented in the readout ASICs. The internal (fixed) veto latency of the example is set to 5 bunch cycles. Source: [41]

to bunch #105, no vetoes have occurred. The addresses buffered in the FIFO are of monotonic order. For the next incoming bunch #106, the storage location would be #106. A veto is issued now at the beginning of bunch #106, causing the counter to pause and hold its present value #106. Considering the value of the fixed latency in this example, the veto counts for bunch #101, which is reflected in the output of the shift register. That is, storage cell #101 is immediately overwritten with the data of bunch #106. Address #101 is (again) buffered in the shift register and could be overwritten if a veto for bunch #106 is received (which happens during bunch #111). The counter proceeds operation, and since no veto is issued in the next cycle, data of bunch #107 are stored in cell #106. The monotonic order is now broken.

The fast control protocol also implements two special commands for testing purposes. The command **START\_TESTPATTERN** causes the ASICs to continuously send test patterns over the serial data links. On the other hand, **STOP\_TESTPATTERN** immediately stops the transmission of test patterns and returns the master FSM into idle state.

## Slow-control

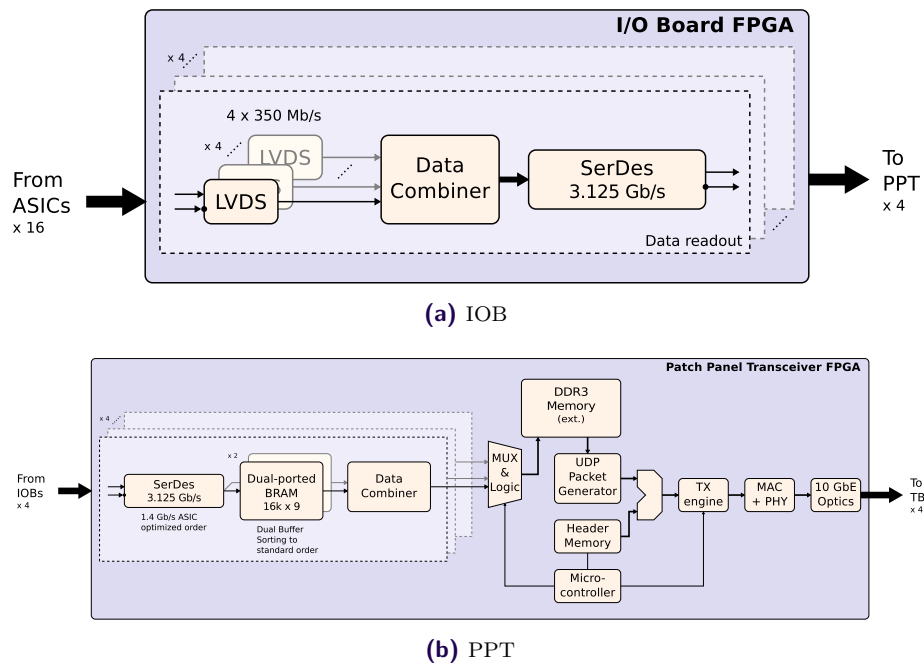
Slow-control in general covers configuration, monitoring, and maintenance of the DSSC system. A remote PC of the individual experiment station runs a custom software framework developed by the software group of the European XFEL. The software communicates with an embedded microcontroller – a Xilinx MicroBlaze soft-core CPU – on the PPT via gigabit Ethernet. The

MicroBlaze will run an ordinary Linux kernel<sup>3</sup> from the standard kernel tree. This combination is a well-established solution for FPGA-based embedded systems that provides a high degree of flexibility. In that way, the microcontroller can easily extend the detector-specific plug-in of the DAQ software on the remote PC.

The IOBs as slave modules are controlled via the PPTs through a bi-directional slow-control interface, which is also used to provide the configuration settings that operate the peripheral electronics of a detector module.

### 5.1.3 Data Transport

Figure 5.4 illustrates the data path of the DSSC DAQ for an IOB (a) and a PPT (b). Data from



**Figure 5.4:** Block diagram of the DSSC DAQ data path, illustrated for an IOB (a) and a PPT (b).

the ASICs of a sensor Main Board arrive at the IOB FPGA over 16 input channels. The data words are 10 bits in size and are transmitted MSBit first at a data rate of 350 Mb/s. The leading bit is a start bit and always '1', while the following nine bits are the actual ADC pixel data<sup>4</sup>. Data from the 16 channels are joined by a data combiner and buffered in a FIFO, before the high-speed transceivers (SerDes, GTPs) of the FPGA transmit them towards the PPT at a speed of 3.125 Gb/s per transmission lane.

At the PPT, the high-speed SerDes (GTX) of the FPGA receive the incoming data of four IOBs, which at this point are arranged according to an ASIC-optimized readout mechanism. As the TB expects pixel-ordered image data provided in UDP packets, a combination of FPGA-internal dual-buffered memory and a data combiner is used to re-order the data to coherent images. The

<sup>3</sup>No special embedded Linux kernel required

<sup>4</sup>For operation at nominal bunch frequency of 4.5 MHz, the ADC provides 8-bit pixel data, while 9-bit data words are foreseen for operation frequencies of less than 2.2 MHz only.

sorted data are copied to external DDR3 memory, which provides storage capacity and bandwidth for processing full images. A UDP packet generator successively loads the frame data from the DDR3 memory and generates a data stream of custom XFEL UDP packets. As the maximum size of UDP packets is 64 kB, the image data must be transmitted in small chunks, for example 4k pixel or 16k pixel. Assuming 16k pixel per UDP packet, the packet rate for which header information must be generated is in the order of 20 kHz. This can be accomplished by software running on an embedded microcontroller – the Xilinx MicroBlaze soft-core CPU – which also monitors receive progress from the IOB and transmission towards the TB. In order to achieve maximum data throughput, both low-level Ethernet packetizing and data transmission are managed by a hardware TX engine, which combines pixel data with the header information generated by the microcontroller. The finalized Ethernet packets proceed to a 10GbE MAC and a 10GbE PHY, before they are transmitted over optical 10 Gb/s transceivers. Incoming UDP packets – if any – will be forwarded to the microcontroller and subsequently be processed by software. However, this is optional and not a functional element of the main data path. Table 5.2 summarizes both native link and payload data bandwidths of the interfaces pixel data have to pass from the ASIC to the TB.

Interface	ASIC → IOB	IOB → PPT	PPT → TB
Native link speed (per device)	350 Mb/s	4 x 3.125 Gb/s	4 x 10 Gb/s
Payload data volume (start bit + 9-bit data)	≈ 26.2 Mb	≈ 419.4 Mb within 99.6 ms	≈ 1.68 Gb
Payload data rate	263.2 Mb/s	4.21 Gb/s	16.84 Gb/s
Connection type	LVDS	GTP	QSFP+ (optical)
Protocol	Custom ASIC	Aurora	Custom UDP/IP

**Table 5.2:** Native and payload data bandwidths of the internal DSSC DAQ interfaces. The numbers are provided on a per-link level of the interfaces.

The data flow presented in this document is based on using all four available high-speed SerDes devices (GTPs/GTXs), which was foreseen in the initial draft of the DSSC DAQ concept. Recent DAQ development, however, implements data transmission only over three transceiver lanes for technical reasons, as the PPT will provide a three-channel high-speed interface per IOB. Considering the limited payload data rate of about 4.2 Gb/s per module, there is no restriction from omitting the fourth high-speed channel.

## 5.2 Hardware Implementation

The PPTs and the IOBs are custom-developed PCBs that make use of latest<sup>5</sup> FPGA and Ethernet technologies to meet the requirements of the DSSC detector readout. For both of the two DAQ elements, FPGA devices from *Xilinx* have been chosen as target platforms<sup>6</sup>. While there are other popular FPGA manufacturers like *Altera* and *Lattice*, which offer devices of similar capability and performance, Xilinx was chosen for reasons of a widespread application in industrial and scientific projects. Xilinx devices also have already been used at ZITI in different scientific applications, for example at ATLAS experiment (high-energy physics experiment at CERN) [34] or

<sup>5</sup>As of the time of elaborating the implementation draft

<sup>6</sup>For naming conventions of Xilinx devices the reader may refer to the documentation of the respective chip family provided at <http://www.xilinx.com/>



for astrophysical simulations [42]. Table 5.3 opposes the decisive characteristics for the FPGAs of different manufacturers taken into consideration.

	Device	Size	I/Os (Banks)	High-speed SerDes (Data rate)	Cost (approx.)
PPT	Altera EP4S360	27 mm × 27 mm	560 (n/a)	16 (8.5 Gb/s)	8,700 \$
	Lattice LFSC25	33 mm × 33 mm	476 (n/a)	16 (3.8 Gb/s)	515 \$
	Xilinx XC7K325T	31 mm × 31 mm	500 (10)	16 (12.5 Gb/s)	1,380 \$
IOB	Altera EP4CGX30	19 mm × 19 mm	150 (9)	4 (2.5 Gb/s)	120 \$
	Lattice EPC3-35	17 mm × 17 mm	133 (7)	4 (3.2 Gb/s)	100 \$
	Xilinx XC6SLX45T	15 mm × 15 mm	310 (4)	4 (3.125 Gb/s)	90 \$

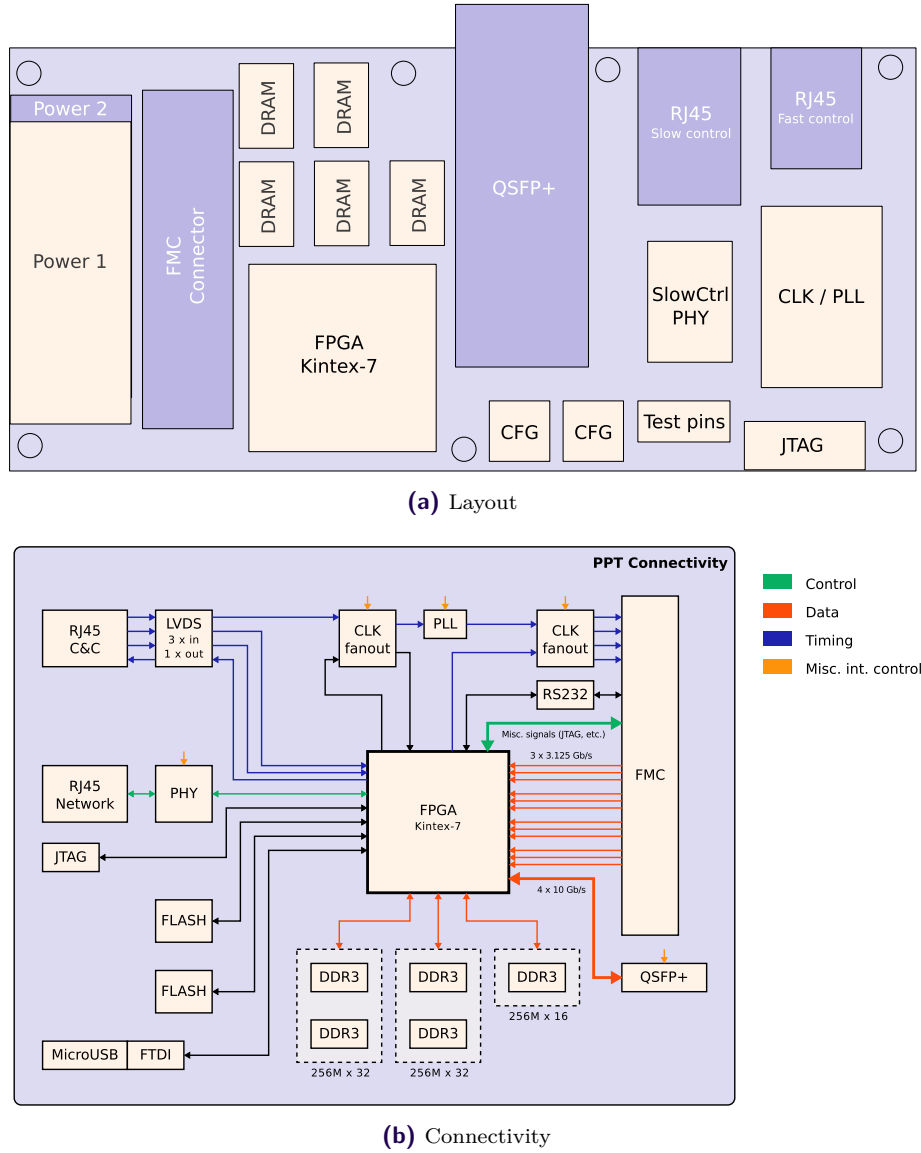
**Table 5.3:** Comparison of different FPGA devices from Altera, Lattice, and Xilinx, taken into account for the layout of the PPT and the IOB.

The PPT and IOB use different families of Xilinx FPGAs according to the needs of their application. A PPT uses a Kintex-7 as central element, which is reasonable since 16 high-speed transceivers at 12.5 Gb/s each are required at least. The PPT device additionally has to manage sophisticated tasks like generating the detector timing, operating on data of the 64 ASICs of a quadrant, and controlling / monitoring the DSSC via gigabit Ethernet slow control. On the other hand, an IOB only has relatively simple tasks to perform, such as controlling peripheral components of the Main Board and capturing the data from 16 ASICs, without further computation on them. The availability of four serial high-speed SerDes devices and also the limited board dimensions due to the detector geometry demand for a device of the Spartan-6 FPGA family.

### 5.2.1 The Patch Panel Transceiver

The conceptual design of the PPT is presented in figure 5.5, which illustrates (a) the layout and (b) the connectivity of the device prototype. The PPT is currently under development at ZITI Mannheim. It may be noted that the implementation presented in the following already employs the reduced number of high-speed transceiver channels between IOB and PPT.

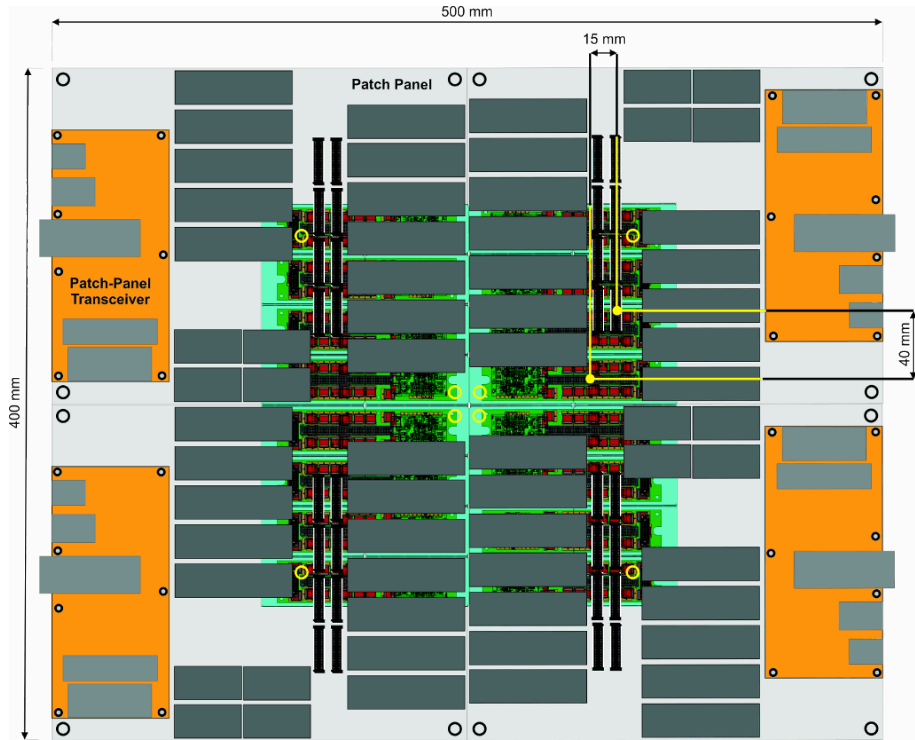
The PPT will be implemented on a FMC-based PCB with a dimension of about 150 mm × 70 mm. As shown in figure 5.6, all four devices will consume only about a quarter of the total DSSC patch panel surface, which allows plugging four the PPT mezzanine cards on top of the patch panel without interfering with other cable connections required for the detector. A Xilinx Kintex-7 XC7K325T-2-FFG900-C FPGA provides 16 high-speed SerDes devices (GTX) capable of up to 12.5 Gb/s per lane, which is sufficient to cover 12 input data links from four IOBs and four output links towards the TB. In addition, the integrated 10GbE PHY directly drives the optical transceivers, which makes an external, dedicated PHY device obsolete and gives room for other components. The optical connection to the TB is realized using QSFP+ transceiver modules like Avago AFBR-79E4Z, which is another step in density optimization. Consequently, an MPO-to-LC breakout cable as shown in figure 5.7 is required to match the SFP+ interface on the TB side. As for the interface to the C&C system, an RJ45 connector (Molex 446610001 or similar) is used to receive and transmit the three LVDS timing signals and the FEE LVDS status signal, respectively. The slow-control gigabit Ethernet interface is implemented with another low-profile RJ45 connector (Belfuse L834-1G1T-S7) and a Micrel KSZ9021RN GbE PHY that connects to an FPGA I/O



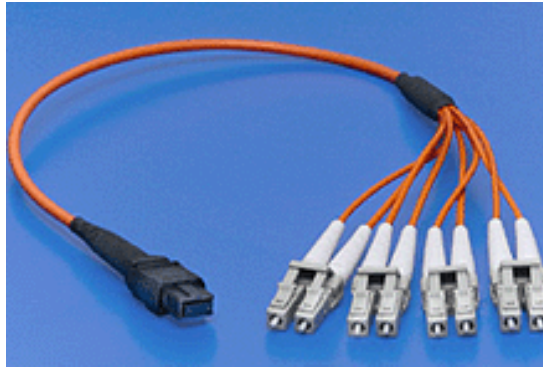
**Figure 5.5:** Layout and connectivity of the PPT prototype. Dark parts reside on the bottom side of the PCB.

bank. Five DDR3 chips of type Hynix H5TQ4G63MFR-H9C or Micron MT41J256M16-125E<sup>7</sup> are arranged in three groups next to the FPGA and serve as buffers for the pixel data of a quadrant. The boot code for FPGA configuration and the microcontroller is stored in two flash memory chips (e.g. S25FL128AGMFI001 and S25FL256SAGMFI001 from Spansion) which can be accessed via a JTAG connector (Molex 87832-1420). Several local clock oscillators (XO) are required to provide the FPGA main clock (200 MHz) and a reference clock for the high-speed transceivers (156.25 MHz). The oscillators will be of type SiLabs SI530. In order to generate the clock of about 695 MHz for the ASIC ADCs from the C&C FEE clock (approx. 99 MHz), a PLL of type ADF4351 (Analog

<sup>7</sup>The two chips run at a clock frequency of 800 MHz max. and feature a 16-bit wide data bus



**Figure 5.6:** Location of the PPT modules (orange) on top of the DSSC patch panel. Only about a quarter of the total patch panel area is occupied by the mezzanine cards.

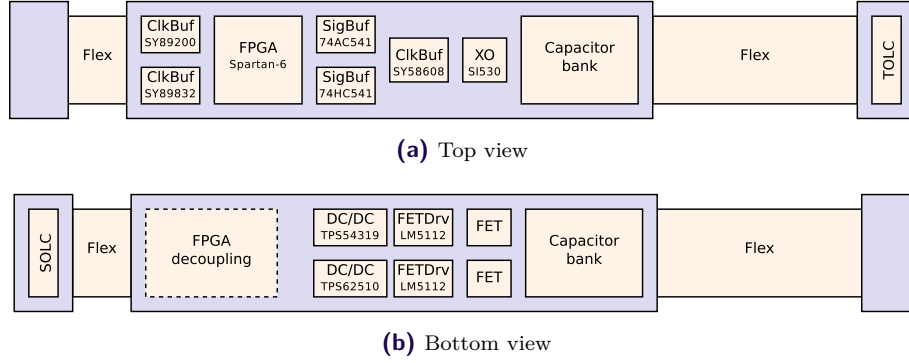


**Figure 5.7:** MPO-to-LC breakout cable as it is required to interface the PPT with the TB.  
Source: [43]

Devices) is used. This device features a typical jitter of 0.3 ps rms [44]. Each PPT board will have a silicon serial number device (Maxim DS2401), which supplies the PCB with a unique ID that is electronically readable.

## 5.2.2 The I/O Board

The IOB has a spatial dimension of approx.  $229.3\text{ mm} \times 22.6\text{ mm}$ , which is identical to that of a PRB. A sketch of the IOB prototype that has been developed within the scope of this thesis is illustrated in figure 5.8. In the following, revision-specific details are distinguished by using the



**Figure 5.8:** Sketch of the IOB layout.

term *IOB-rev0.1* for the prototype version, while *IOB-rev1.0* refers to the revised version of the board that will most likely be applied in the final DSSC. If no specific details need to be stressed, IOB is used as a general term.

The central element of the IOB-rev0.1 is a Xilinx Spartan-6 XC6SLX45T-3-CSG324-C FPGA. The main reasons for choosing this device are the number of featured high-speed transceivers and the physical size of the chip. An LX45T offers four GTPs at 3.125 Gb/s each, which is sufficient for the expected data rate of approx. 4.21 Gb/s of a sensor module. The CSG324 package with  $15\text{ mm} \times 15\text{ mm}$  is the smallest size available for an LX45T and allows a dense implementation on the PCB. For the prototype, a device that features commercial temperature range ( $0^\circ\text{C}$  to  $+85^\circ\text{C}$ ) is used due to the fact that the initial test system will be operated at room temperature only. Moreover, the difference in electrical characteristics between commercial and industrial ( $-40^\circ\text{C}$  to  $+100^\circ\text{C}$ ) temperature range is negligible according to the Spartan-6 data sheet [45]. Speed grade -3 has a few advantages over -2, mainly related to the minimum and maximum ratings of the FPGA clock management specifications. A speed grade of -1 drops out, since there is no LXT device available in this category. Since the DSSC system will be cooled down to approx.  $-20^\circ\text{C}$ , the IOB-rev1.0 will be supplied with an XC6SLX45T device of speed grade -3 featuring an industrial temperature range.

Figure 5.9 illustrates the connectivity of the IOB FPGA. A 60-pin terminal strip of type TOLC (115-02-L-Q-A) and a 60-pin socket strip of type SOLC (115-02-L-Q-A) (both from Samtec) connect the IOB to the MIB and the Main Board, respectively. Contrary to the PPT, the IOB has no flash storage for the firmware. Instead, the FPGA fabric is configured via a JTAG interface hosted by the PPT. A slow-control interface provides the inter-communication between the PPT and the IOB, which is used to send commands for remotely control the IOB and its peripheral electronics.

Two different sets of control signals from the IOB FPGA are provided to the PRBs using octal signal drivers of type TI CD74HC541PW and Fairchild 74AC541MTC, respectively. The first device operates the shift registers that define the status of the ASIC power nets. The latter one controls gate drivers that generate the clear pulses for the DEPFET sensor.

The 1-to-8 fanout clock buffers (National Semiconductor LMK01010) residing on the Main Board directly connect to the IOB FPGA via a serial interface. The interface provides access to internal

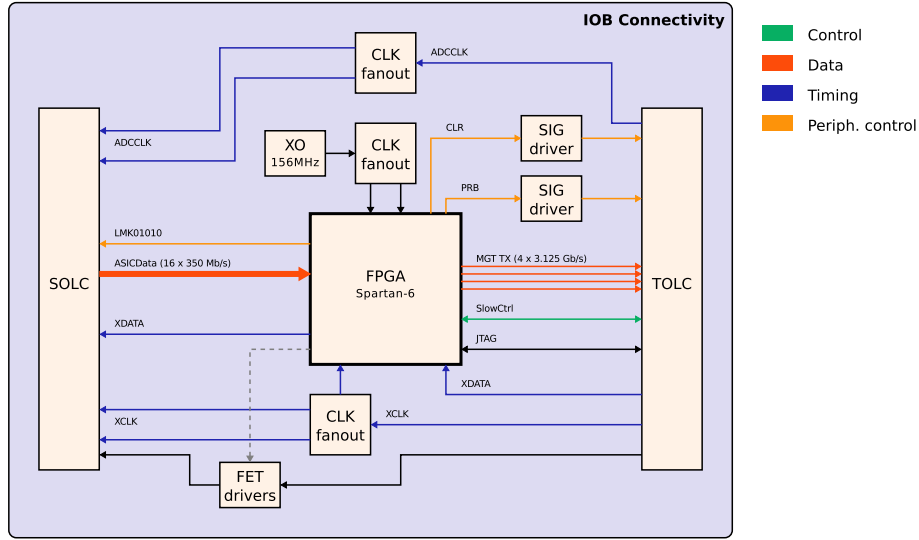


Figure 5.9: Connectivity of the IOB FPGA.

configuration registers, which for example allow adjusting the input-to-output delay of the clock signal. This feature is essential to compensate for signal skew, as the clock signals distributed by the LMK devices must have a well-defined phase relation to the electron bunch timing.

The clock signals used as ADC sampling clock and for ASIC fast-control are received two ultra-precision fanout clock buffers from Micrel. An SY89200U (1-to-8) and an SY89832U (1-to-4) each duplicates one of the input clocks **ADCCLK** and **XCLK** and distributes a copy to the clock buffers of the two Main Board sections. A third copy of each clock signal is provided to the IOB FPGA for purposes of synchronizing and data recording. As the Spartan-6 specification permits a maximum clock input frequency of 400 MHz only, the frequency of **ADCCLK** signal is divided by two<sup>8</sup> before feeding into the Spartan-6. Since the ASICs send the pixel data synchronously to half the frequency of **ADCCLK**, there is no downside of doing so.

The ASIC data are received over signal lines that directly link the SOLC to an FPGA I/O bank. After de-serializing inside the FPGA, data are forwarded to the PPT via the four (three<sup>9</sup>) GTP lanes and the TOLC.

The power for the sensors is received over the TOLC and supplied via the SOLC. For reasons of minimizing the power consumption during the readout phase, the power is gated by connecting the corresponding nets to a set of FETs. The IOB FPGA operates the FETs by means of LM5112MY FET drivers from National Semiconductor.

An XO of type SiLabs SI530FA156M25DG generates the local clock for the FPGA. The clock signal is duplicated by a Micrel SY58608U 1-to-2 fanout buffer, which provides the copies to the FPGA as fabric clock and separate reference clock for the high-speed transceivers.

As for the local power required by the IOB electronics, a set of DC/DC switching converters derive additional supply voltages from the 3.3 V main power net supplied over the TOLC. Two TI TPS62510 devices produce separate 1.2 V power networks, which are used as FPGA core / I/O supply voltage and for powering the GTP high-speed transceivers. A single TI TPS54319 device generates the 2.5 V supply voltage for another FPGA I/O bank and various peripheral electronic components.

<sup>8</sup>In the final system it might also be divided by four and multiplied accordingly inside the FPGA fabric

<sup>9</sup>Latest PPT concept foresees only three links.

## 5.3 Connectivity of the DSSC DAQ

Figure 5.10 visualizes the overall DAQ connectivity of a detector quadrant.

## 5.4 Comparison of the Front-end DAQ Systems

In the previous and present chapter, the concepts and implementations for the front-end DAQs of AGIPD, LPD, and DSSC have been introduced. The outstanding requirements on the detectors regarding timing and data readout demand for readout systems with low-latency and high data bandwidth. As a consequence, all three detector consortia concluded to use custom FPGA-based DAQ components, which represent the best solution considering efficiency and flexibility with reasonable development effort.

Although global prerequisites like a common timing system and optical 10GbE interfaces for data transmission are given, the different detector specifications (resulting from different research goals) account for a system-specific development of the front-end DAQ for each individual detector type.

The concept of the DSSC DAQ system is based on a master-slave implementation at detector quadrant level. The PPT as a master provides the interfaces to the XFEL back-end DAQ and is responsible for global timing and control of, and data transmission from a detector quadrant. The IOB primarily handles data acquisition and low-level controlling tasks of peripheral electronics on a sensor Main Board level.

The hardware realization of the DSSC DAQ system is driven by the idea of providing a maximum of efficiency and flexibility by using modern technologies. Using Xilinx FPGA devices of both Spartan-6 and Kintex-7 families has a number of advantages over prior device families. The most significant one is the fact that both devices provide I/O capabilities that sufficiently cover the bandwidth requirements for using only one FPGA per DAQ module, which yields to a much denser implementation.

In contrast to the DAQ systems at AGIPD and LPD, which provide storage capacity for about 350 and 512 bunches, respectively, the DSSC DAQ is capable of recording more than 640 images during a train. Using devices from latest Xilinx FPGA families allows minimizing the total number of FPGAs applied within the entire DSSC DAQ to a total number of  $20^{10}$ . For the LPD system, the total number of FPGAs is  $64^{11}$ , and for AGIPD it is  $84^{12}$ . Both AGIPD and LPD DAQ systems make use of an additional 10GbE FMC (with dedicated PHY device) for optically interfacing with the TB, whereas the Kintex-7 device of a DSSC PPT incorporates a 10G PHY that can directly drive the optical transceivers. Other features such as the QSFP+ transceiver modules allow for further optimized implementation of the DSSC DAQ with regard to the area consumed by electronic components.

From the technological point view, the DSSC front-end DAQ system is an elegant solution that benefits from latest technology in micro electronics wherever possible.

---

<sup>10</sup>16 IOBs with 1 FPGA per IOB and 4 PPTs with 1 FPGA per PPT

<sup>11</sup>16 FEM with 4 FPGAs per FEM

<sup>12</sup>4 control boards with 1 FPGA each and 16 readout boards with 5 FPGAs each

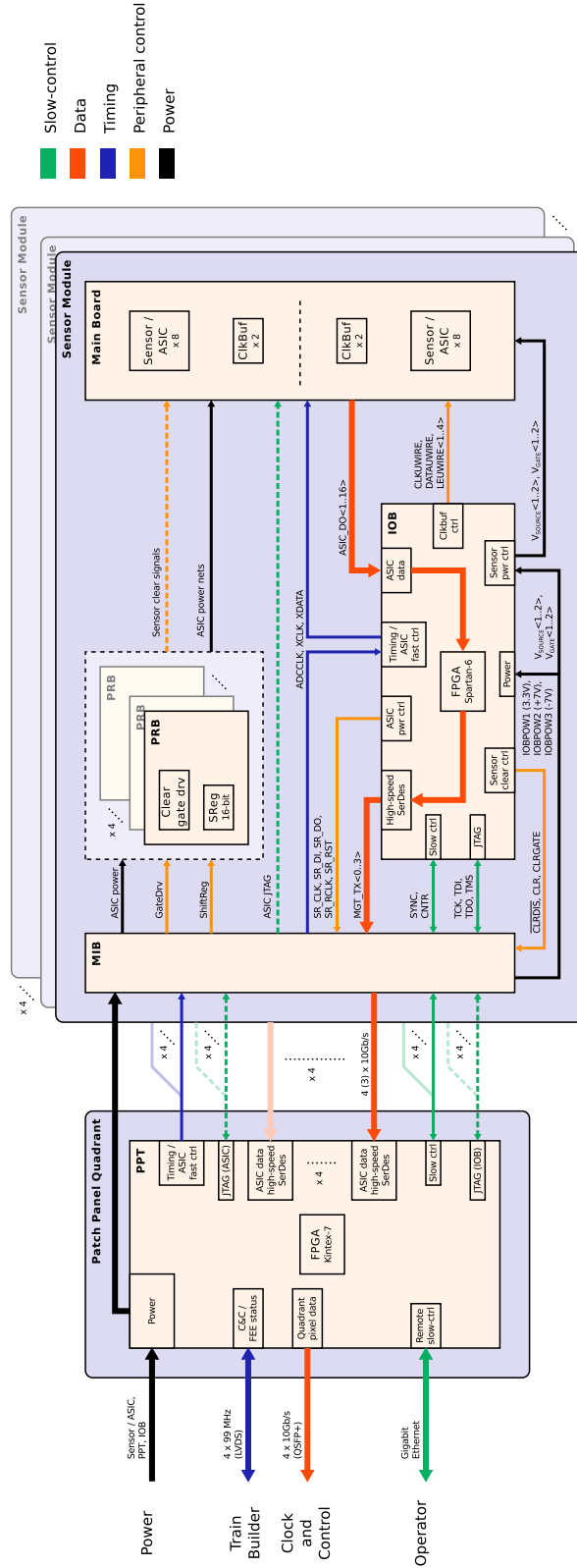


Figure 5.10: Overall DAQ connectivity of a DSSC quadrant.

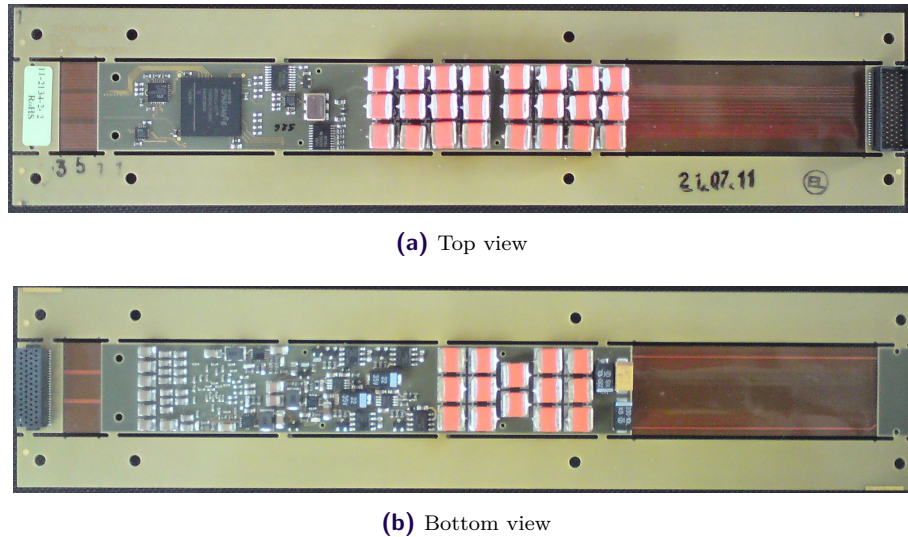




## 6 The I/O Board Prototype

The concept of the DSSC DAQ system follows a master-slave approach with two FPGA-based sub-components, the PPT and the IOB. The IOB is responsible for low-level control of peripheral detector module electronics, but also for reading out the digital ASIC pixel data during the inter-train gap and providing them to the PPT via high-speed data channels. As the detector geometry accounts for a dense implementation of the IOB, special considerations regarding PCB layer stack-up and trace routing were elaborated in order to ensure signal integrity particularly for the critical high-speed signals.

Figure 6.1 shows a photograph of the IOB-rev0.1 prototype, whose development was one major subject of this thesis.



**Figure 6.1:** Top and bottom view of the IOB-rev0.1 prototype (with PCB framing). During the soldering process, the Polyimide core of the long flexible tail became partially delaminated, as it can be seen in picture (b). The results of the electrical tests, however, confirm that the delamination has no degrading effect on the signal transmission characteristics.

As the signal naming slightly changed with regard to IOB schematics and the further detector development, the table given in chapter A.2 lists both initial and actual names of the signals for a convenient understanding.

The first part of this chapter summarizes fundamental considerations on signal integrity that are of importance in high-speed PCB design, and also describes the electrical interfaces of the IOB. The second part focuses on IOB-specific topics such as realization of the PCB layer stack-up, but also on special circuitries which perform dedicated tasks like switching the power of both the sensors and the ASICs. A summary of the modifications and improvements incorporated into the design for the revised IOB version is presented in the last part of this chapter.

## 6.1 Design Considerations

The process of designing a PCB usually accounts for applying a number of fundamental design rules. Particularly, PCBs which provide high-speed signalling demand for careful and well-considered routing to meet the requirements for a good signal integrity. For the design of the IOB, several other prerequisites like interfacing to external electronics and FPGA I/O bank assignment have to be taken into account.

### 6.1.1 Signal Integrity

Signal integrity is of major concern at all high-speed PCB traces. Precautions against different noise-inducing aspects like reflection and crosstalk, and also minimizing power and ground noise must be taken into consideration to ensure a reliable signal transmission. The considerations presented in this section are mainly founded on [46, 47], which should be consulted for a deeper understanding of the theory of signal integrity.

#### Reflection

Noise induced by reflection can cause overshoot, undershoot, and ringing in high-speed systems. Avoiding reflections represents one of the most demanding parts in signal integrity-driven designs. Reflection noise is caused by impedance discontinuities along the signal transmission path, which can be reduced by proper signal termination, impedance-matched traces, and a minimum number of vias.

**Termination** Without proper termination, reflections at either end of a high-speed trace superpose the source signal and consequently add noise to it. Generally, one distinguishes between source termination at the transmitter side and end termination on the receiver side.

For source termination, each driving gate is connected through a series resistor to its transmission line. The sum of output impedance of the driver plus resistor value should equal the target impedance of the transmission line, thus the intensity of the propagating signal is half of the source signal from the driver. On the receiving side, the half-sized reflection plus the half-sized source signal bring the signal to a full level. The reflection propagates back to the source termination resistance, where it is eventually damped.

End-terminated transmission lines provide a resistor at the receiving end of the signal trace, which damps all reflections. The value of the terminating resistance should match the line impedance, which is also the target impedance. As a consequence, the propagating signal is of same intensity as the source signal from the driver. Additionally, end termination has the advantage over source termination of providing the possibility for daisy-chaining multiple receivers.

**Impedance Matching** The impedance of a trace depends on the geometric dimension of the copper trace (i.e. width and thickness), but also on parameters like the dielectric constant of the surrounding material and the distance to the nearest conducting reference layer(s). The impedance of a trace is typically calculated by using a field resolver algorithm, which is provided with the parameters for trace geometry and insulator material. While solving the complex algorithm by

hand can be a very demanding task, modern PCB design software provides tools that interactively allow calculating the trace parameters to match a specific impedance.

## Ground Bounce and Power Supply Noise

Ground bounce refers to shifts in the internal ground reference voltage due to output switching, which induces noise on the signal lines. Precautions that minimize the effect of ground bounce are to provide low-inductance of ground connections, which can be realized by using large-area ground planes. Also, splitting the ground planes and interconnecting the slices by small electrical bridges can reduce the effect of ground bounce, as the signal return paths are decoupled, while still keeping them on a common potential.

Stable supply voltages are of major significance in digital designs, since they are used by the electrical components as references to interpret the state of a received input signal (i.e. "low" or "high"). Noise on the reference signal narrows the margin in which a proper state decision can be made.

Generally, power supply noise originates from two sources. One of them is the high-frequency switching noise produced by some components (e.g. clock drivers) applied in digital designs. If such components connect to common power and ground lines, the generated noise spreads across the entire power network. *Decoupling* isolates two circuits on a common line by adding a low-pass filter (usually a ferrite bead and a capacitor) between the supply input of the circuit and the global power net. The low-pass filter attenuates the high-frequency content of any current that passes through the ferrite bead into the global power net.

The inductance of voltage regulators and their wiring causes the time-varying current demand of a connected load to produce high-frequency noise on the supply voltage. For reducing this noise voltage, one can lower the wiring inductance and reduce the change of rate of current flowing through the inductor. The latter is accomplished in *bypassing* the high-impedance signal path with capacitors which represent a low-impedance shunt for the varying load currents. Using a combination of parallel capacitors of different values allows for suppressing several noise frequency bands. Lowering the wiring inductance usually is achieved by increasing the mutual coupling of the supply wire to its return path using power planes along with a ground plane. In placing the planes very closely to each other, the capacitance increases, while at the same time the impedance decreases for high current frequencies.

## Differential Signalling

Differential signalling solves integrity issues that occur for single-ended transmission lines at high frequencies and long-distance connections. On differential traces, the signal is transmitted over two transmission lines of complementary polarity. The information is encoded in the difference between the two signal levels. As a consequence, a differentially transmitted signal is far less sensitive to the noise sources such as cross talk or power / ground noise. The traces of a signal pair are affected equally, and the resulting common mode noise is filtered by the differential receiver.

On the other hand, noise caused by reflection is still an issue. Since the traces of a differential pair are closely coupled, avoiding impedance discontinuities is an essential measure, not only with regard to the single line impedance of the individual traces, but also with respect to the differential impedance between the two lines.

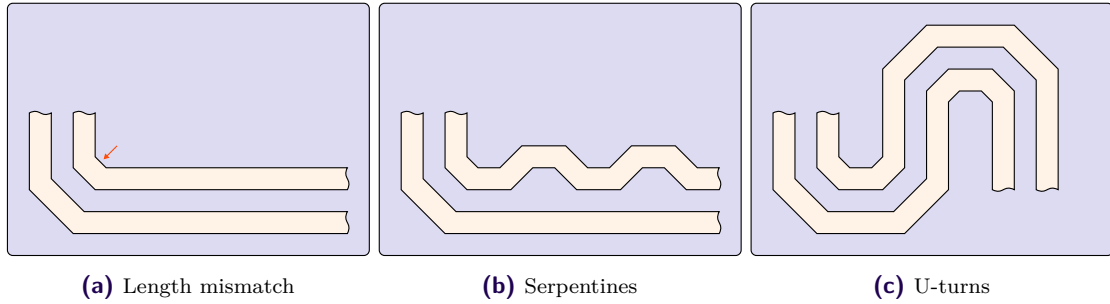
Inside the layer stack-up of a PCB, differential pairs can be arranged either horizontally within the same layer (edge-coupled) or vertically on two adjacent layers (broad side-coupled). Generally,

the manufacturing tolerances in the position of traces within a layer are smaller than in the layer height. Thus, horizontally aligned differential traces have a more precise matched impedance than vertically aligned ones.

**Length Matching** Proper routing of differential high-speed signal traces implies two significant considerations:

- Achieving length matching for optimum timing margins and preventing common-mode signals
- Maintaining constant separation between the traces of a pair for matching the target impedance

When the two conditions can not be satisfied simultaneously, meeting the first should be preferred over the second. The primary purpose of length matching is to keep jitter to a minimum. The bendings that occur when a differential pair is routed often involve a length mismatch between the two traces, as shown in figure 6.2 (a). A difference in the trace lengths directly translates into a



**Figure 6.2:** Length matching of differential traces. While serpentine are used to align the individual traces of a differential pair, u-turns allow for length matching different trace pairs to each other.

delay between the signals on the positive and negative line, according to:

$$\Delta t_{sig} = \Delta t_{propdelay} \cdot \Delta L \quad (6.1)$$

where  $\Delta t_{sig}$  is the signal delay,  $\Delta t_{propdelay} = \frac{1}{v}$  is the propagation delay given by the reciprocal of the signal velocity  $v$ , and  $\Delta L$  is the difference between the trace lengths. A typical approximation for the signal velocity inside an FR4<sup>1</sup> PCB is  $v = 152.4 \frac{\text{mm}}{\text{ns}}$ , which corresponds to about  $t_{propdelay} = 6.56 \frac{\text{ps}}{\text{mm}}$  [48].

That is, even a fraction of a millimeter induces a jitter in the order of a few picoseconds, which has a negative effect on high-speed signal integrity.

With careful routing techniques, the length mismatch of a differential pair can be minimized or even avoided. One option is to increase the length of the shorter trace by adding small serpentine, as depicted in figure 6.2 (b). However, this technique results in discontinuities in the trace separation, which affects the impedance of the trace. The other option is to route the differential pair with an equal number of u-turns, as shown in figure 6.2 (c). The latter technique is used when matching two differential trace pairs to each other, which is preferable if data transmission is realized over multiple parallel channels. Generally, both techniques are applied in practice.

<sup>1</sup>Glass-reinforced epoxy laminate commonly used as standard base material for production of PCBs.

### 6.1.2 Interfaces and Signals to External Electronics

The IOB has to provide a variety of signals to the outside, for both its local power supply and the interfacing to peripheral electronics. These signals are provided through both the 60-pin SOLC and TOLC connectors at each end of the board.

**JTAG** The fabric of the IOB FPGA requires an interface that allows the device being configured remotely. Xilinx FPGAs offer several configuration interfaces [49], of which JTAG has been chosen due to its convenient operation mechanism. The four interfacing signals TCK (clock), TDI (data-in), TDO (data-out), and TMS (mode-select) are provided through the TOLC connector. The optional reset signal TRST is omitted.

**Slow-control** Remote control of the IOB by the PPT is realized with a dedicated slow-control interface. It is implemented as two-wire<sup>2</sup> serial interface with the signals SYNCCLK (differential clock) and CNTR (single-ended bi-directional data) and uses three pins on the TOLC. The clock signal is of 2.5 V LVDS I/O standard, while the data signal uses 1.2 V LVC MOS.

**ASIC Data Channels** Each of the 16 readout ASICs of a Main Board directly interfaces its 2.5 V LVDS data lane ASIC\_D0<n> (where n refers to the ASIC number between 1 and 16) with the IOB FPGA. In total, a number of 32 pins is required on the SOLC for connecting all readout chips.

**High-speed Transceivers** The IOB FPGA features four high-speed transceivers, which are used for transmitting the recorded ASIC pixel data towards the PPT. The four differential lanes MGT\_TX<n> (with n between 0 and 3) are of 1.2 V CML I/O standard and require eight pins on the TOLC.

**ASIC Control Signals** The 2.5 V LVDS fast-control signals ADCCLK, XCLK, and XDATA for the readout chips use six pins on the TOLC, but 10 pins on the SOLC, since some of the signals are provided twofold, one for each Main Board section. The global reset line XRESET requires one pin on both the TOLC and SOLC and is of 1.2 V LVC MOS I/O standard.

**Clear Signal Gate Driver Control** The gate drivers that generate the DEPFET clear signals and reside on each PRB require three 3.3 V LVC MOS control signals  $\overline{\text{CLRDIS}}$ , CLR, and CLRGATE for being operated by the IOB FPGA. The signals are provided over three pins of the TOLC.

**Clock Buffer Programming** The four programmable fanout clock buffers located on the sensor Main Board each provide a three-wire 3.3 V LVC MOS communication interface that is connected to the IOB FPGA and allows configuring the devices. In particular, the input-to-output delay of the clock signal can be adjusted for the individual device outputs in order to compensate for skews in the signal path. For minimizing the required pin count, the signals for both clock CLKuWire and data DATAuWire are distributed to each device in parallel. The third signal serves as latch-enable and is distributed separately for each clock buffer over LEuWire<n> (with n from 1 to 4). A total number of six pins is used on the SOLC.

<sup>2</sup>Actually three wires, since the clock is differentially transmitted.

**Analog / Digital ASIC Power Control** The analog and digital power required by the readout chips can be selectively switched off during the readout phase for reasons of minimizing power consumption. The shift registers residing on the PRBs hold the bitmasks that define the state of the power nets. The interface to the shift registers is realized by the signals `SR_CLK` (clock), `SR_DI` (serial data-input of shift registers), `SR_RCLK` (latch enable), `SR_RST` (reset), and `SR_DO` (data-out from shift registers). The signals use 3.3 V LVC MOS as I/O standard and require five pins on the TOLC.

**Sensor Power** The main power for the DEPFET sensors is provided over the IOB from external power supplies to each Main Board section. The three power nets `VGATE<n>` (3.3 V..7.3 V), `VSOURCE<n>` (4 V..7 V), and `VSSS<n>` (GND<sup>3</sup>) (where n refers to the Main Board section number and is 1 or 2) use eight pins on the TOLC as well as on the SOLC.

**Local Power** The active electronics of the IOB requires different supply voltages, of which not all can be provided from the external power supplies due to pin count limitations. Only the main power (3.3 V) is supplied via four TOLC pins on the power net `IOBPOW1`. The other local supply voltages (2.5 V and 1.2 V) are derived from the main power net using DC/DC converters. Another two power nets `IOBPOW2` (5 V..7 V) and `IOBPOW3` (−5 V..−3 V) are used for operation of the sensor power switching circuitries and require two pins on the TOLC.

### 6.1.3 FPGA I/O Assignment

The Spartan-6 LX45T FPGA provides four I/O banks which are used for interfacing with the peripheral electronics. Since different I/O standards are required, the banks are provided with different supply voltages.

**Bank Config ( $V_{CC} = 2.5\text{ V}$ )** provides dedicated pins for configuration of the FPGA fabric and connects to the four-pin JTAG interface with `TCK`, `TDI`, `TDO`, and `TMS`.

**Bank 0 ( $V_{CC} = 2.5\text{ V}$ )** is solely used for feeding the LVDS reference input clocks into the global clock tree of the FPGA (`MAINCLK`, `REFCLKGTP`, `REFCLK400`, `XCLK`, and `SYNCCLK`).

**Bank 1 ( $V_{CC} = 1.2\text{ V}$ )** interfaces with the LVC MOS signals `CNTR` (slow-control data) and `XRESET` (global detector reset line).

**Bank 2 ( $V_{CC} = 2.5\text{ V}$ )** provides the 2.5 V LVDS interface to both data and control signals of the readout chips. In particular, the 16 differential data channels `ASIC_DO<1..16>` and the ASIC fast-command signal `XDATA` are connected to this I/O bank.

**Bank 3 ( $V_{CC} = 3.3\text{ V}$ )** interfaces with the required LVC MOS signals at 3.3 V. These are all signals for the Main Board fanout clock buffers (`*uWire`), the PRB shift registers (`SR_*`), and the clear signal gate drivers (`CLR_*`). The signals that control the on-board FET drivers of the sensor power switching circuitries (`V*_CTRL`) are also connected to bank 3.

**Bank 101 / 123 ( $V_{CC} = 1.2\text{ V}$ )** are the dedicated banks of the high-speed transceivers (GTPs). The high-speed interface is unidirectional, that is, only the transmitter lanes `MGT_TX<0..4>` are used, while the RX channels are grounded. The GTP reference clock `REFCLKGTP` is provided through bank 101.

---

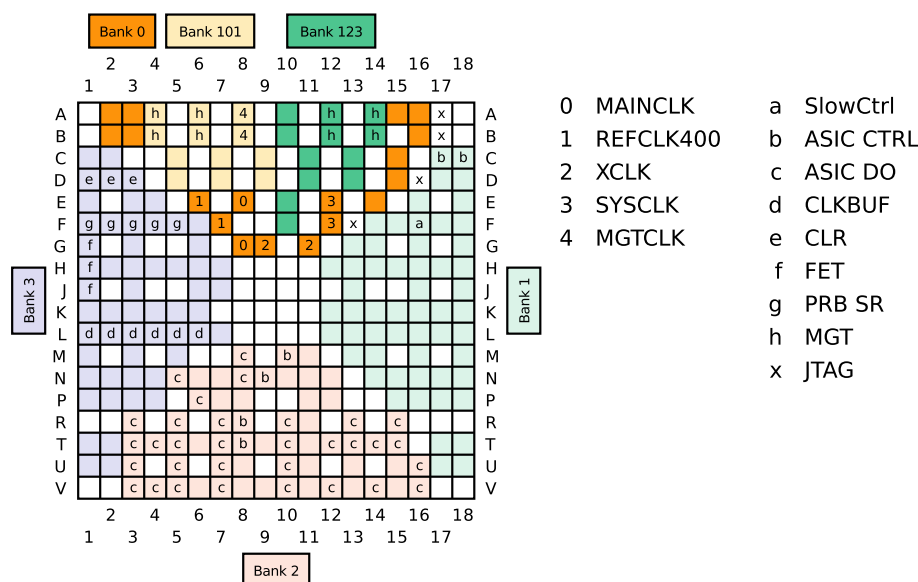
<sup>3</sup>Initially foreseen as −7 V..−5 V, but changed to GND for noise performance reasons of the sensor

Table 6.1 summarizes the I/O bank utilization of the FPGA device. The pinout and the I/O bank

Bank	Total pins	Pins used	$V_{CC}$	I/O standard	Usage
Config	8	4	2.5 V	LVC MOS	JTAG
0	18	8	2.5 V	LVDS	Reference input clocks
1	56	3	1.2 V	LVC MOS	Slow-control data, global detector reset
2	60	36	2.5 V	LVDS	ASIC data / control
3	56	17	3.3 V	LVC MOS	Miscellaneous peripheral electronics
101	12	6	1.2 V	LVDS / CML	High-speed transceiver (clock, TX)
123	12	4	1.2 V	CML	High-speed transceiver (TX)

**Table 6.1:** I/O bank usage of the IOB FPGA. In total, three different I/O standards at three different I/O voltages are used. The usage numbers show that only about 35 % of the available I/O pins are used.

occupation of the Xilinx Spartan-6 LX45T device are illustrated in figure 6.3.



**Figure 6.3:** Utilization of Spartan-6 LX45T FPGA device.

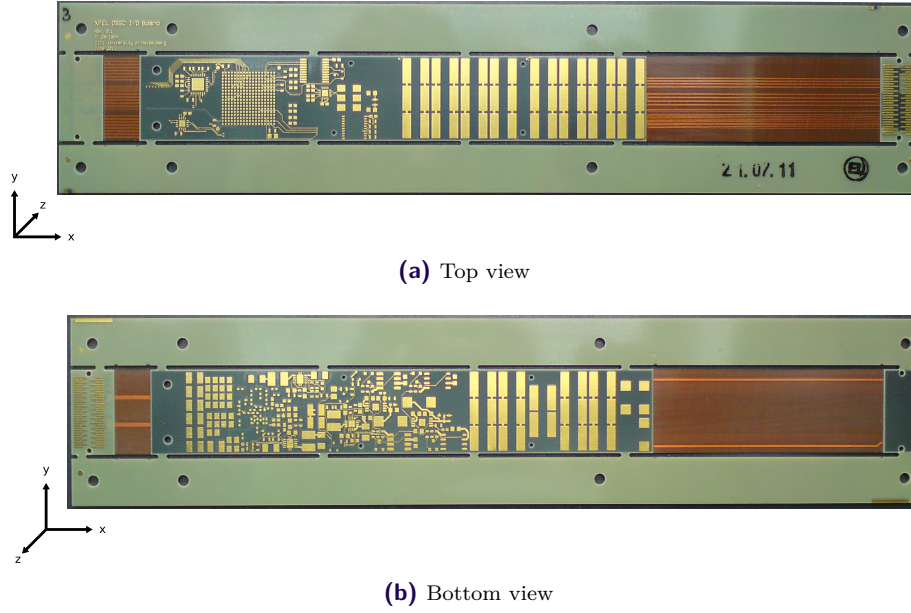
A number of considerations have to be taken into account when assigning the I/O resources of the FPGA. I/O, reference and termination voltages have to comply to a set of banking rules, which are specified by the vendor. Further constraints for the assignment are fixed clock region, special pins, and simultaneous switching output (SSO) limitations. A detailed description of the related constraints can be found in the Spartan-6 User Guides at [50].

## 6.2 PCB Layout

The IOB PCB has a physical size of 229.3 mm × 22.6 mm with a total thickness of approx. 1.4 mm. Figure 6.4 shows a photograph of the top (a) and bottom (b) side of the IOB-rev0.1 blank (with



framing). In horizontal ( $x$ -) direction, the board is divided into three rigid sections. The IOB is



**Figure 6.4:** Top (a) and bottom (b) side of the IOB-rev0.1 blank with framing. The large quadratic arrangement of pads on the middle rigid section marks the location of the FPGA. The SOLC and TOLC land patterns are placed on the outer left (bottom) and outer right (top) parts, respectively.

realized as a 10-layer PCB ( $z$ -axis) with a flexible lead core that runs across the entire board and interconnects the rigid parts. The middle section houses all active IOB electronics, while the two outer sections are used for mounting the SOLC and TOLC only. The flexible core is required for mechanical reasons. Both SOLC and TOLC are available as straight connectors only. Since the IOBs have to be mounted perpendicularly onto the sensor Main Board, a bending connection with a flexible lead has been pursued. In the initial draft of the DSSC detector mechanics, the long flex region towards the TOLC was foreseen to provide the movability of the individual detector quadrants to form the hole for dumping the primary XFEL beam. In the present realization, the moving capability of a quadrant is provided through another flexible interconnection between the detector quadrant plate and the feed-through plate / PPT. The key parameters of the IOB PCB are summarized in table 6.2.

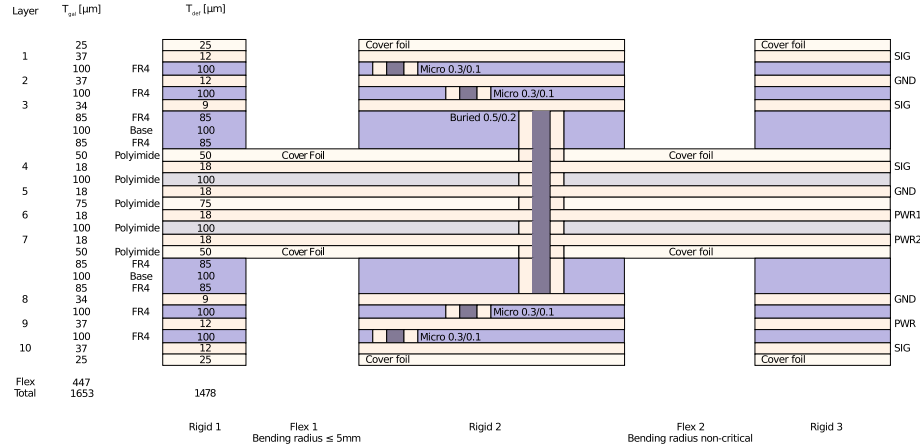
Dimension	229.3 mm $\times$ 22.6 mm
Total thickness	1.403 mm
No. of layers	10
No. of flex layers	4
Length of electronics rigid	135 mm
Length of flex lead	63 mm
Length of flex bend	9.8 mm
Bending radius	$\approx$ 5 mm

**Table 6.2:** Key parameters of the IOB PCB.



### 6.2.1 PCB Layer Stack-up

The IOB PCB layer stack-up shown in figure 6.5 demonstrates the utilization and thickness of the individual layers. The four innermost layers are based on a flexible *Polyimide* core, on which three



**Figure 6.5:** Layer stack-up of the IOB PCB. The PCB provides a flexible four-layer Polyimide core, on which the remaining six layers are symmetrically stacked upon. The signal transport layers are shielded from the dedicated power layers by ground planes. Micro-via technology is used to interconnect the layers. However, the flex core requires special treatment of the interconnecting via, since single-layer connections are not possible due to the production process.

subsequent layers of standard FR4 epoxy laminate are glued on both the top and bottom side. Both top and bottom layers (L1 and L10, respectively) are used for component placement and mixed signal routing. The second layer L2 is a ground plane and serves as reference plane for layer one and three. L3 is a signal layer primarily used for the distribution of differential high-speed signals, but also for some non-critical single-ended signals. The adjacent flex core provides the main transport layers for signalling, power distribution, and grounding. Layer L4 is solely used for signal traces (both single-ended and differential) and represents the only signal transport layer between the two outer rigid parts and the middle electronics section. The main ground plane is formed by layer L5, which is the reference for L4 and additionally shields it against noise from the two power layers L6 and L7, respectively. L5 is designed as a split ground plane and is additionally used for distributing the global detector reset line. L6 supplies the power required by the local IOB electronics from the TOLC to the middle rigid section. The sensor power nets are explicitly routed on L6 and L7. Leaving the Polyimide core, layer L8 represents another ground layer that shields the sensor power layers against layer L9, which represents the main distribution layer for the local supply voltages derived from the IOB main power. Similar to the ground layers, L9 is designed as a split power plane.

The critical high-speed signals are mainly routed on L3 and L4 due to improved shielding and well-known impedance. Only a few short distances of the impedance-controlled differential traces are routed on the surface layers L1 and L10.

The interconnection between the individual layers is realized with *micro-via* technology using both stacked and staggered buried vias. The vias are filled with copper and plugged in order to reduce their electrical resistance. While the layer pairs (L1,L2) and (L2,L3) as well as (L8,L9) and (L9,L10) can be interconnected individually through single buried micro-vias, the Polyimide core does not support single-layer interconnections. Instead, vias that cut through all four flex layers plus one

adjacent rigid layer on both sides are required. Moreover, the via diameter has to be increased as well for robust through-connections. Table 6.3 summarizes the properties of the applied layer stack-up and micro-vias.

Insulator material	FR4
Insulator dielectric constant	4.2 . 4.5
Flex material	Polyimide
Flex dielectric constant	3.7
Minimum trace width	80 $\mu\text{m}$
Minimum trace spacing	80 $\mu\text{m}$
Differential traces	Edge-coupled
Via technology	Micro-via (buried, stacked, and staggered)
Via size (mm)	0.3/0.1 (FR4), 0.5/0.2 (polyimide)

**Table 6.3:** Properties of the IOB layer stack-up and the applied vias. The insulator material of the rigid layers is standard FR4, while for the flex core it is Polyimide.

## 6.2.2 Split Power and Ground Planes

The IOB uses several dedicated split power and ground planes. While splitting the power planes (L6 and L9) mainly serves the purpose to reduce the number of power layers within the stack-up, the more essential splitting of the ground planes (L2, L5, and L8) minimizes the induced ground noise by decoupling dedicated signal return paths. This is of particular significance for the transmission lines of the high-speed transceivers. Table 6.4 lists the different split ground planes and their associated signals. However, the ground planes are not totally separated from each other. Small bridges –

Ground Plane	Associated Signals	Description
MGT_GND	MGT_TX*	Explicit GTP ground
GATE*_N	GATE<1..2>_P, VSOURCE*, VSSS*	Return path for sensor power nets
GND_M	CLRDIS, CLR, CLRGATE	Sensor clear signal ground reference
IOBGND	IOBPOW<1..3>	Return path for IOB power nets
ADCGND	All other signals	Ground reference for all remaining signals

**Table 6.4:** Split ground planes and their associated signals applied on the IOB prototype.

realized by short copper traces and micro-vias – interconnect the split planes and keep them on constant ground potential to prevent ground shifting.

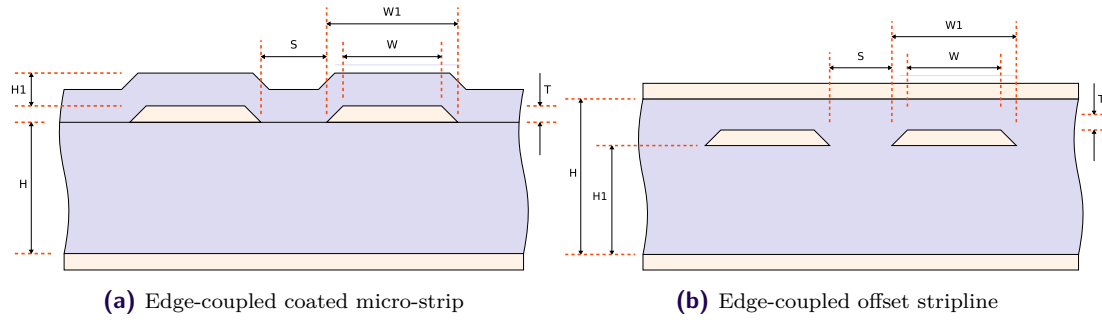
## 6.2.3 Impedance Calculation

Differential and single-ended high-speed signals are typically matched for a target impedance of approx. 100  $\Omega$  and 50  $\Omega$ , respectively. However, the high-speed interfaces of the IOB are strictly realized using differential signalling, and the single-ended traces are not impedance-controlled at all. Table 6.5 lists the parameters that are used for calculation to match the target impedance of the individual IOB signal layers as close as possible. As a general rule, the spacing between the traces of a differential pair should not be much larger than twice the individual trace width

Layer	Area	Type	H	H1	W	W1	S	T	$\epsilon_r$	$Z_0$
1	rigid	Edge-coupled coated micro-strip	100	25	90	100	210	37	4.5	99.55
3	rigid	Edge-coupled offset stripline	572	438	70	80	200	34	4.2	100.98
4	rigid	Edge-coupled offset stripline	572	100	90	100	130	18	3.7	99.15
	flex	Edge-coupled coated micro-strip	100	25	90	100	90	18	3.7	100.24
10	rigid	Edge-coupled coated micro-strip	100	25	90	100	210	37	4.5	99.55

**Table 6.5:** Impedances of the differential traces on the IOB, including trace geometry parameters used for impedance calculation.

to achieve a close coupling. A graphical representation of the meaning of the trace parameters is depicted in figure 6.6.



**Figure 6.6:** Illustration of the trace parameters for (a) edge-coupled micro-strip and (b) edge-coupled offset stripline.

### 6.2.4 Length Matching

The critical differential high-speed signals on the IOB-rev0.1 are length-matched to minimize the delay between the positive and negative signal fraction. Both u-turn and serpentine techniques are applied. In particular, the high-speed transceiver lanes have a maximum inner-pair tolerance of 0.01 mm. The maximum difference in the lengths of two transceiver lanes is in the order of 0.07 mm. The ASIC data signal traces on the other hand show a inner-pair mismatch of 0.15 mm to 2.26 mm. The spatial limitations on L4 prevent length-matching measures of these signals. However, the deviation is still within the common recommendation of 2.54 mm (100 mil). Table 6.6 summarizes the signal names and the deviation from matching.

Signal	MGT_TX	ADCCLK	ASIC_D0	XCLK	XDATA	MAINCLK	REFCLKGTP
$f$ (approx.)	3.125 GHz	695 MHz	347.5 MHz	99 MHz	99 MHz	156 MHz	156 MHz
$\Delta L$ (max.)	0.01 mm	0.04 mm	2.26 mm	0.01 mm	0.37 mm	1.96 mm	0.64 mm

**Table 6.6:** Summary of the trace length tolerances of the differential high-speed signals.

## 6.2.5 Signal Termination

Termination of the differential high-speed signals is realized internally by the corresponding receiver devices (end-terminated), that is, no explicit termination resistors are applied. Since the single-ended traces are not impedance-controlled, they do not require any termination precautions.

## 6.2.6 Power Bypassing

The bypassing network for the IOB serves the purpose to minimize the noise on the power supply signal paths and contains about 70 bypass capacitors of different types and values. A list of the different capacitors applied is given in table 6.7. The majority of them belongs to the decoupling

$\begin{array}{c} \text{[}\mu\text{F]} \\ \text{[V]} \end{array}$	0.1	0.22	0.47	1	4.7	100	220	Total
1.2	-	8	16	-	12	3	-	39
2.5	-	-	11	-	5	3	-	19
3.3	2	-	4	-	2	1	1	10
7.0 . 9.0	-	-	-	-	-	-	1	1
-5.0 . -3.0	-	-	-	2	-	-	1	3

**Table 6.7:** Capacitor utilization for the bypassing network of the IOB power supply. The first column and row represent the supply voltage and the capacitance values, respectively. The number within a cell represents the number of capacitors.

network for the Spartan-6 FPGA [51], while the other active electronics mainly uses single bypass capacitors.

## 6.3 Special Circuitries

The IOB electronics comprises a number of dedicated circuitries that perform specific tasks. In particular, circuitries for providing the local supply voltages, controlling the sensor power switching, and generating the sensor clear control signals have been elaborated and are presented in the following.

### 6.3.1 Local Power Supplies

**Estimated power dissipation** For selecting suitable supplies that provide a sufficient amount of power to the IOB electronics, it is essential to estimate the total power dissipation which is expected during operation. For an accurate estimation, not only the static, but also the dynamic power and currents must be considered. Generally, the data sheet of an electronic device provides an estimation of both static and dynamic current flow and power consumption, respectively. For an FPGA, however, the current flow and the power dissipation strongly depend on factors like I/O bank utilization and I/O standards being used. Thus, the *Xilinx Power Estimator (XPE)*<sup>4</sup> has been used to assess power dissipation for the Spartan-6 device. Table 6.8 gives a per-device

Supply voltage [V]	Device	Power [mW]	
		Per device	Total
1.2	LX45T-CSG324-3-C	790	790
2.5	LX45T-CSG324-3-C	300	1640
	SY89200	890	
	SY89832	260	
	SY58608	190	
3.3	LX45T-CSG324-3-C	< 10	750
	74AC541	250	
	74HC541	220	
	TPS54319	110	
	TPS62510 (2x)	50	
	LM5112 (2x)	30	
Total power dissipation			approx. 3180

**Table 6.8:** Estimation of IOB power dissipation.

overview of the estimated power dissipation for the IOB. Additionally, an analysis of the firmware power dissipation was done using the *Xilinx Power Analyzer (XPA)* tool. The tool calculates the required power by analyzing the firmware logic. The results of both XPE and XPA are summarized in chapter A.1.

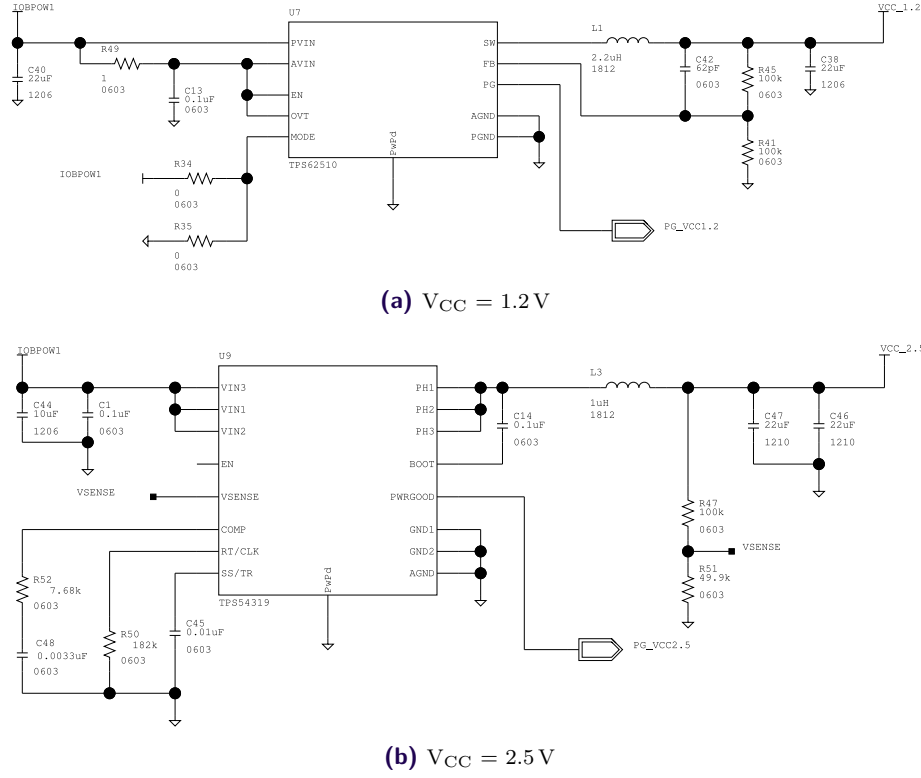
Figure 6.7 shows the schematics for the 1.2 V (a) and 2.5 V (b) local power supply circuitries. The circuitries are based on TPS62510 and TPS54139 DC/DC switching converters from Texas Instruments, respectively, which generate the supply voltages from the 3.3 V main power net. The specification of the devices comply with the requirements on the supply current as estimated for the calculation of the IOB power dissipation. Xilinx recommends a separate 1.2 V power supply solely used for powering the serial high-speed transceivers [52], which is realized by another identical TPS62510 circuitry on the IOB.

Both output voltage and switching frequency of the converters are determined by configuring their peripheral electronic components accordingly. The values of the passive components are calculated using the formulas given in the device data sheets at [53, 54].

### 6.3.2 Sensor Power Switching

In order to further minimize the overall power dissipation of the detector system, the DEPFET sensors are powered only during the macro-bunch period and switched off during the inter-train gap. More precisely, the sensor supply voltages  $V_{\text{SOURCE}}$  and  $V_{\text{GATE}}$  are gated by dedicated FET-based switching circuitries. The switching mechanism is controlled by the IOB FPGA synchronously to the XFEL bunch timing. Figure 6.8 (a) illustrates the schematic of the switching circuitry for  $V_{\text{SOURCE}}$ . It is based on p-channel and n-channel power MOSFETs of type FDS6681Z and FDN327N (Fairchild), respectively. The devices are capable to handle the requirements on the supply voltage for the DEPFET source contacts, which must be within a range of 4.0 V to 7.0 V (4.0 V typ.) and at supply currents of up to 7 A. The FETs additionally feature a low  $R_{\text{DS(on)}}$ , which minimizes the voltage drop between input and output power nets. A National Semiconductor

<sup>4</sup>[http://www.xilinx.com/products/design\\_tools/logic\\_design/xpe.htm](http://www.xilinx.com/products/design_tools/logic_design/xpe.htm)



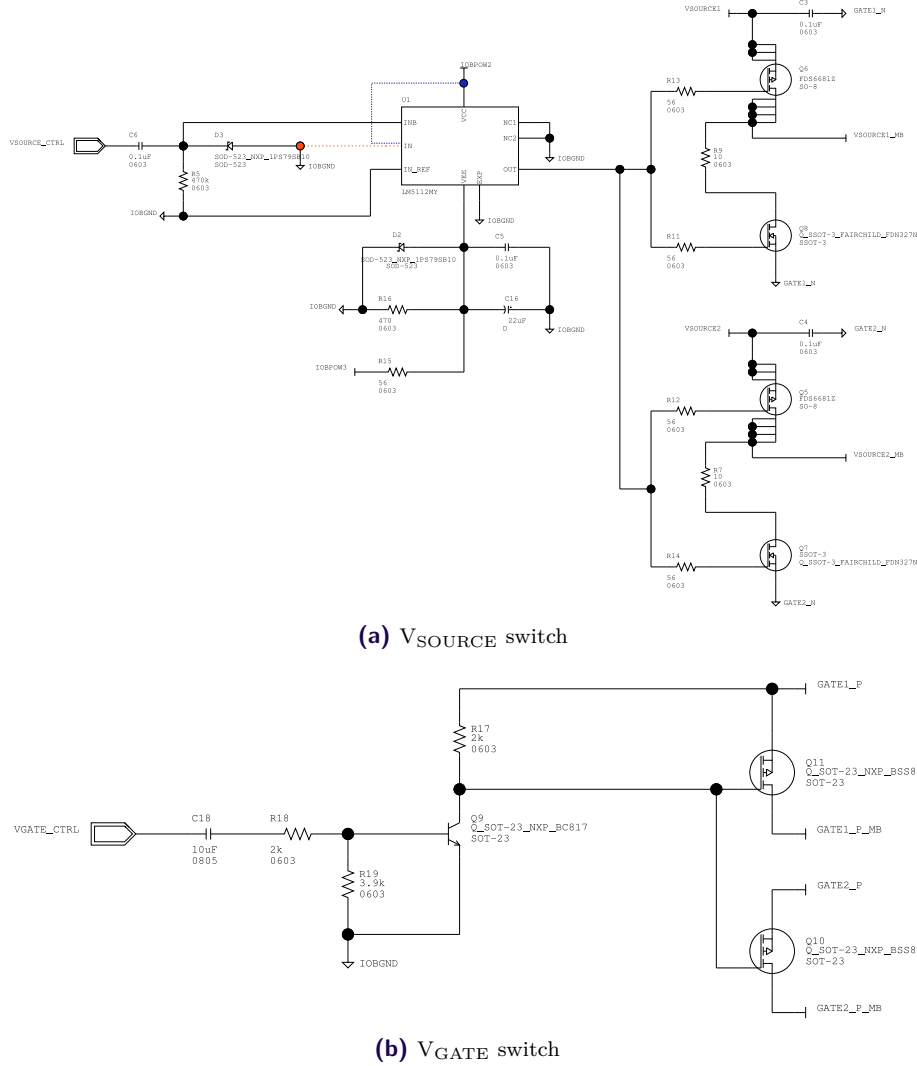
**Figure 6.7:** Schematics of the power supply circuitries providing (a)  $V_{CC} = 1.2V$  and (b)  $V_{CC} = 2.5V$ . The circuitries are realized with a TPS62510 and a TPS54139 switching DC/DC converter from Texas Instruments, respectively. A second identical 1.2 V TPS62510 circuitry is explicitly used for powering the GTPs.

LM5112MY FET driver controls the conductance of both power FETs by steering their gate contacts.

The FDS6681Z p-MOS is placed in the path between  $V_{SOURCE*}$  and  $V_{SOURCE*\_MB}$ , which connects to the corresponding sensor power net on the Main Board. On the other hand, the FDN327N n-MOS is located between  $V_{SOURCE*\_MB}$  and its ground reference  $GATE*\_N$ . In case of a low LM5112MY output signal state, the p-MOS transistor blocks, while the n-MOS transistor conducts and pulls  $V_{SOURCE*\_MB}$  to ground. A high output signal level from the FET driver closes the n-MOS and opens and the p-MOS instead, which supplies  $V_{SOURCE*\_MB}$  with the source voltage  $V_{SOURCE}$ . The resistor between the drain contacts of both the p-MOS and the n-MOS FET limits the cross current during the switching phase that occurs due to a mismatch in the transistor characteristics. The p-MOS / n-MOS arrangement is provided twice, one for each Main Board section.

The input-to-output voltage drop can be further reduced by applying a negative gate voltage to the FETs. Thus, the circuit of LM5112MY is designed for both GND and negative reference level provided via power net IOBPOW3. An additional RC circuit at the FET driver control input limits maximum pulse width and serves as a safety measure to avoid damaging the DEPFET sensor.

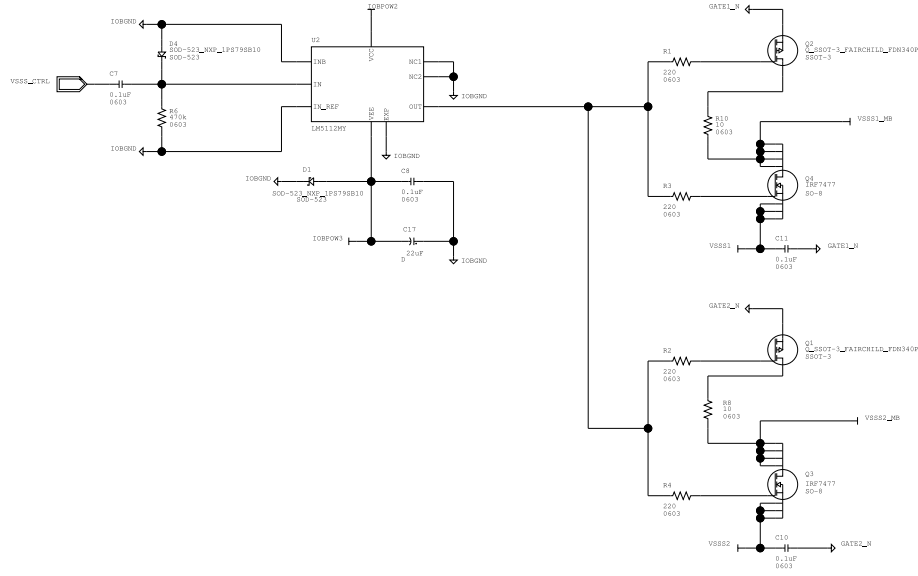
The supply voltage  $V_{GATE}$  for the DEPFET gate contacts must be in the range between 3.3 V to 7.3 V (5.0 V typ.). The nominal flowing current, however, is in the order of a few hundred milliamperes only, thus the gating circuitry can be realized with a BC817-type BJT and two BSS84-type MOSFETs in open drain configuration (one for the each Main Board section), as shown



**Figure 6.8:** The FET-based switching circuitries for (a)  $V_{SOURCE}$  and (b)  $V_{GATE}$  sensor power nets. The bug of a wrongly wired LM5112MY FET driver (red) is manually fixed on the prototype (blue), and is also corrected in the new IOB design.

in figure 6.8 (b).

The IOB prototype supplies another circuitry shown separately in figure 6.9, which was intended for switching  $V_{SSS}$ . Due to the new sensor concept which omits  $V_{SSS}$  as supply voltage, but refers to it as ground potential, the third circuitry has been removed from the design for IOB-rev1.0. The circuitry for  $V_{SSS}$  basically resembles the  $V_{SOURCE}$  circuitry with small modifications. In its initial draft,  $V_{SSS}$  was foreseen for a voltage range from  $-7V$  to  $-3V$ , referred to  $GATE*_N$  as ground level. As a result, the role of p-MOS and n-MOS are swapped, since the (technical) current flows from ground to the negative voltage. Additionally, different MOSFETs (Fairchild FDN340P as p-MOS and International Rectifier IRF7477 as n-MOS) are chosen, which are more suitable for gating negative voltages.



**Figure 6.9:** Obsolete circuitry for switching  $V_{SSS}$  sensor supply voltage.

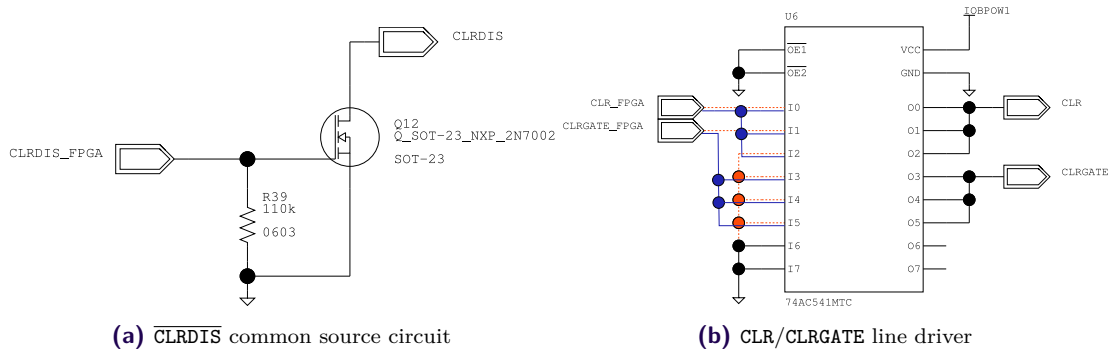
In order to reduce the voltage drop on both  $V_{SOURCE}$  and  $V_{SSS}$  during on-switching, each power net is decoupled with huge capacitance provided by a capacitor bank. For  $V_{SOURCE}$ , a total capacitance of 24 mF (12 x Vishay 597D 1 mF) per voltage group is provided, while for  $V_{SSS}$  it is about 4.7 mF (7 x Vishay 597D 680 uF) per voltage group. Additionally, all three circuitries have RC-coupled control signals as precautions against constantly enabled FPGA signals. The maximum pulse width is limited by the RC time constant  $\tau = R \cdot C$ , which has yet to be determined experimentally for optimum behavior.

### 6.3.3 Sensor Clear Signals

The sensor clear signal are used to empty the drain (CLR) and gate (CLRGATE) contacts of the DEPFET after each bunch (i.e. at a frequency of 4.5 MHz). The characteristics of the DEPFET sensor require a typical signal level of 6.0 V and 19.0 V for CLR low and high, respectively, while for CLRGATE it is in the order of 4.0 V and 12.0 V. Considering the high voltage levels, it is reasonable that careful signal handling is essential for the safety of the DEPFET sensor. Moreover, the FPGA is incapable of providing signal levels this high. As a consequence, special gate drivers are used to generate the clear signals.

The gate drivers reside on the PRBs and are operated by the IOB FPGA with regard to the bunch electron timing via three control signals. For electrical reasons such as drive strength and rise time, the control signals are supplied to the gate drivers via small intermediate circuitries, as depicted in the schematics in figure 6.10. A common source circuit with an n-channel FET generates the low-active control signal  $\overline{\text{CLRDIS}}$ , which disables and enables the clear gate drivers. On PRB side, a pull-up resistor pulls this signal to 5 V. When the control signal  $\text{CLRDIS\_FPGA}$  is asserted, the n-channel FET gets conductive and pulls  $\overline{\text{CLRDIS}}$  to GND, thus enabling the gate drivers. The pull-down resistor at the front of the FET gate contact avoids a floating control signal in case of an unconfigured FPGA fabric, since uncontrolled activation of the gate drivers could severely damage the sensor.





**Figure 6.10:** Schematics of the control circuits for the DEPFET sensor clear signal gate drivers. A common source circuit (a) generates the  $\overline{\text{CLRDIS}}$  signal from an FPGA control signal. The signals  $\text{CLR}$  and  $\text{CLRGATE}$  are supplied through a dedicated line driver (b). The incorrect wiring of the line driver (red) is manually fixed on the IOB prototype and corrected in the IOB-rev1.0 design (blue).

As the minimum guaranteed output current provided by the Spartan-6 device is 24 mA which is not sufficient to drive the  $50\Omega$ -terminated clear control signal lines at a level of 3.3 V, a 74AC541-type line driver was added in the signal path of both  $\text{CLR}$  and  $\text{CLRGATE}$ .

## 6.4 I/O Board Rev. 1.0

The tests that have been carried out on the IOB-rev0.1 prototype have concluded a number of changes and improvements, which are incorporated in the design of the successor revision of the IOB. The following paragraphs briefly summarize these modifications.

**FPGA for industrial temperature range** The FPGA used on the IOB-rev0.1 prototype is designed for commercial temperature range only and is not applicable for the final detector system. The IOB-rev1.0 will be equipped with the same device type, but for an industrial temperature range to fulfill the requirements of the detector operating conditions.

**Removal of obsolete  $V_{\text{SSS}}$  circuit** The control circuit for  $V_{\text{SSS}}$  is entirely removed, since the latest DSSC generation uses  $V_{\text{SSS}}$  as ground reference for  $V_{\text{SOURCE}}$ . The capacitor bank formerly assigned to  $V_{\text{SSS}}$  on the bottom PCB layer is re-assigned to  $V_{\text{SOURCE}}$ .

**$V_{\text{SOURCE}}$  bypass capacitors** The remaining capacitors of the  $V_{\text{SSS}}$  network are assigned to  $V_{\text{SOURCE}}$ . Also, the mounting strategy of the  $V_{\text{SOURCE}}$  capacitors is altered significantly. While on the IOB-rev0.1 prototype, the second layer of capacitors is directly stacked and soldered onto the first, the IOB-rev1.0 provides a plug connection for a dedicated piggy-back board that carries the second capacitor layer on its top PCB layer. In that way, the total capacitance for a  $V_{\text{SOURCE}}$  voltage group is increased from 24 mF to 28.44 mF.

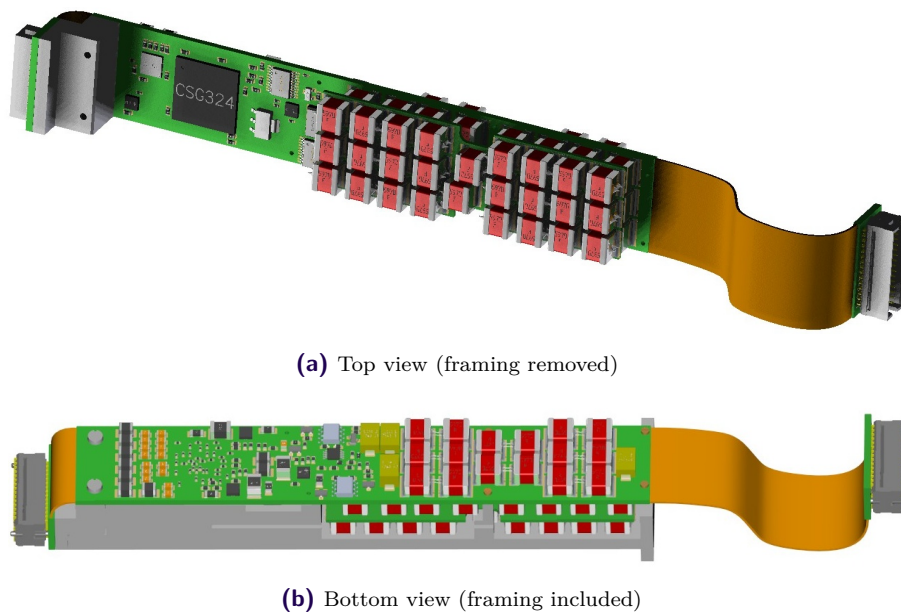
**Removal of GTP AC coupling** The results of the GTP tests with the prototype show a better signal quality when the AC coupling of the high-speed transceiver lanes is replaced by  $0\,\Omega$  resistor bridges. Since the high-speed transceivers of both the IOB and the PPT FPGA operate at the same common mode voltage, the AC coupling of the lanes is omitted.

**Unique Board ID** Each IOB-rev1.0 is equipped with a unique identification number through a *silicon serial number* device of type Maxim DS2401. This allows the operator to keep track of the devices in use. In addition, monitoring and communication can be performed selectively, i.e. slow-control commands can be broadcasted but only be accepted by a specific board.

**Other improvements** A few other improvements are incorporated in the IOB-rev1.0 design. These include

- Minor bug fixes (remapping of some FPGA I/O control signals, fixing incorrect wiring of supply voltage of  $V_{\text{SOURCE}}$  FET driver)
- Removal of LED circuits indicating power condition, which frees area
- Selectable clock divider for the sampling clock provided to FPGA I/O SerDes which capture the ASIC data

Figure 6.11 shows different views of a 3d model of the revised IOB-rev1.0. At the time of writing



**Figure 6.11:** Three-dimensional visualization of the IOB-rev1.0 device.

this document, the PCB is still in production.

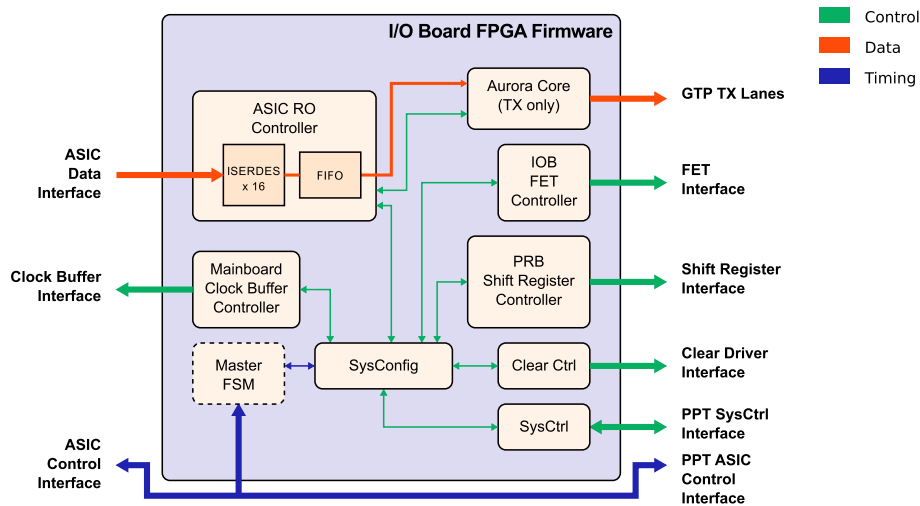
## 7 The I/O Board FPGA Firmware

A major advantage of using FPGA-based DAQ electronics – as opposed to a straight ASIC implementation – is the flexibility provided by those devices. The reconfigurable hardware allows the system to develop with the requirements on the DSSC detector system without changing the hardware layout. By modifying the FPGA firmware, the implemented logic can be adapted to changes in the electron bunch timing structure<sup>1</sup> if necessary, for example. Also, as future applications may demand for new acquisition features, the capabilities of the readout system can be extended by upgrading the firmware.

The first part of this chapter introduces the general concept of the IOB firmware and also presents an elaborate description of the individual firmware modules and their VHDL implementation. The second part describes the test setup that has been used for electrical and functional verification of the IOB and its firmware.

### 7.1 VHDL Firmware

The firmware for the IOB FPGA is composed from a number of VHSIC Hardware Description Language (VHDL) modules, which have not only been developed for testing the IOB prototype, but also for application in the final detector system. A schematic representation of the firmware design is illustrated in figure 7.1. The firmware can be divided into four categories.



**Figure 7.1:** Block diagram of the IOB firmware concept.

<sup>1</sup>Future operation plans foresee increasing the train repetition rate from 10 Hz to 30 Hz.

**Local Timing and Clocking Structure** The timing for operating the IOB is derived locally from the global detector timing distributed by the PPT. In particular, the IOB *master FSM* implements a *timing generator*, which responds to the timing telegrams and triggers controlling processes accordingly. A centralized clock management module, the *Clock Generator (ClkGen)*, implements the clocking structure required by the FPGA logic. In total, the FPGA handles 12 different clock domains, including the XFEL FEE clock (99 MHz), the ASIC data SerDes clock (350 MHz), and the high-speed transceiver reference clock (156 MHz). Synchronization stages are supplied in the control paths of each controller module for proper clock domain crossing.

**System Configuration and Control** Various configuration data of the system are stored in a centralized register bank, which is implemented by the *System Configuration (SysConfig)* module. The *System Controller (SysCtrl)* represents the slave module of the IOB remote slow-control mechanism and provides the PPT with both read and write access to the SysConfig.

**Peripheral Control** Several dedicated controlling entities manage the configuration and control of peripheral electronics which the IOB interfaces with. The *Power Regulator Board Controller (PrbCtrl)* and the *Clear Controller (ClrCtrl)* modules operate the shift registers and clear signal gate drivers located on the PRB, respectively. The FET drivers, which gate the sensor supply voltages  $V_{\text{GATE}}$  and  $V_{\text{SOURCE}}$ , are controlled by the *FET Controller (FetCtrl)*. The programming of the fanout clock buffers residing on the Main Board is handled by the *LMK01010 Controller (LmkCtrl)* entity. The controllers retrieve their configuration data from the SysConfig register bank.

**Data Transport** The data path for ASIC readout is realized with two main modules. The *ASIC Readout Controller (AsicRoCtrl)* captures and de-serializes the incoming data by means of the FPGA IOSERDES mechanism. The controller additionally provides logic that packetizes the data and buffers them in a FIFO. The *Aurora core* from Xilinx collects the data from the FIFO and transmits them over the GTPs using the Aurora protocol. The data transport is exclusively managed by the master FSM.

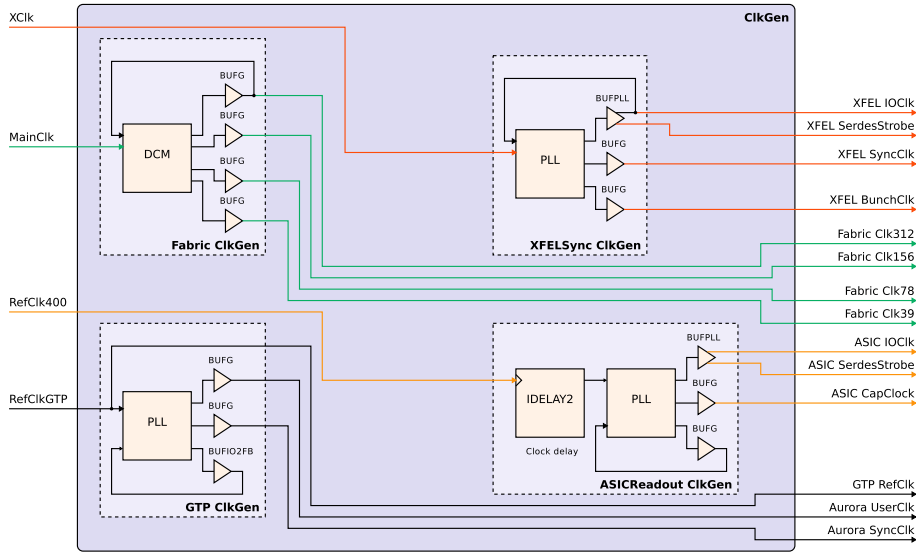
Most of the modules are based on a generic VHDL code implementation to provide a maximum of flexibility and scalability with regard to various test setups, in which the number of peripheral components may differ, or they are not present at all. By adjusting code-specific constants, the modules can be configured for different test environments without rewriting the entire code.

### 7.1.1 Clocking Structure and Local Timing

The various fabric clock domains used by the controller modules are generated by the ClkGen module from both on-board and externally provided reference clocks. The IOB master FSM is responsible for the local sequences that operate the controllers synchronously to the electron bunch timing.

#### Clock Generation

The block diagram in figure 7.2 illustrates the concept of the clock generation module. The FPGA receives four input clock signals from different sources, which are provided to corresponding sub-modules for clock generation. Signal **MainClk** (156.25 MHz) is the FPGA fabric main clock.



**Figure 7.2:** Schematic of the central clock generator module ClkGen.

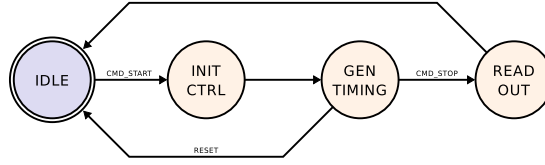
Most of the controllers are driven by this clock or a derivative generated by the *Fabric ClkGen* sub-module. A dedicated reference clock for the GTPs is provided by **RefClkGTP** (156.25 MHz). Sub-module *GTP ClkGen* generates GTP-specific clocks from this clock signal. The XFEL FEE clock **XClk** (99 MHz) serves as input clock to the *XFELSync ClkGen* sub-module. The module produces bunch-synchronous clock domains, which are required by the IOB master FSM for local timing generation. The ASIC data capturing clock **RefClk400**<sup>2</sup> (350 MHz) is provided as reference input clock to *ASICReadout ClkGen*. The sub-module derives the clock signals required by the SerDes mechanism, e.g. the high-speed I/O clock, which is a multiple of **RefClk400** according to the SerDes ratio<sup>3</sup> specified in the VHDL code. A fifth input clock signal serves as transmission clock for the slow-control interface and directly connects to the SysCtrl module. As no other clocks are derived from this signal **SyncClk**, it is not shown in this picture.

## Master FSM and Timing

The master FSM of the IOB FPGA firmware will serve the purpose to implement the control logic to operate each controller module synchronously with regard to the electron bunch timing. The PPT broadcasts control telegrams to the ASICs, which are also monitored by the IOB master FSM. Accordingly, the FSM will enable the controller modules and initialize peripheral electronics. A possible transition graph for the IOB master FSM is illustrated in figure 7.3. It basically consists of the four states **IDLE**, **INIT\_CTRL**, **GEN\_TIMING**, and **READ\_OUT**. Reception of a start command triggers a transition from the idle state to the initialization state. This state could set up several counters, which are used for internal bunch counting and generation of timing. Bunch counting can be of particular interest with regard to different bunch patterns and also the vetoing mechanism, as it provides a measure to double-check the veto information stored on PPT side. After initialization, the FSM enters the state responsible for generating the bunch-synchronous local timing. With the

<sup>2</sup>A reference clock of 400 MHz was initially aimed for when the IOB development has been started.

<sup>3</sup>The SerDes ratio is the ratio between the internal fabric clock used to provide the parallel data input vector, and the high-speed I/O clock used to output the serial data stream.



**Figure 7.3:** Simplified draft of a possible transition graph for the IOB master FSM.

reception of a stop command, the FSM starts the readout of the ASIC data and eventually returns to idle state after completion.

At the time of writing this thesis, a draft version of the IOB master FSM has been implemented which is not fully operational and was thus used for initial testing purposes only.

### 7.1.2 System Configuration and Control

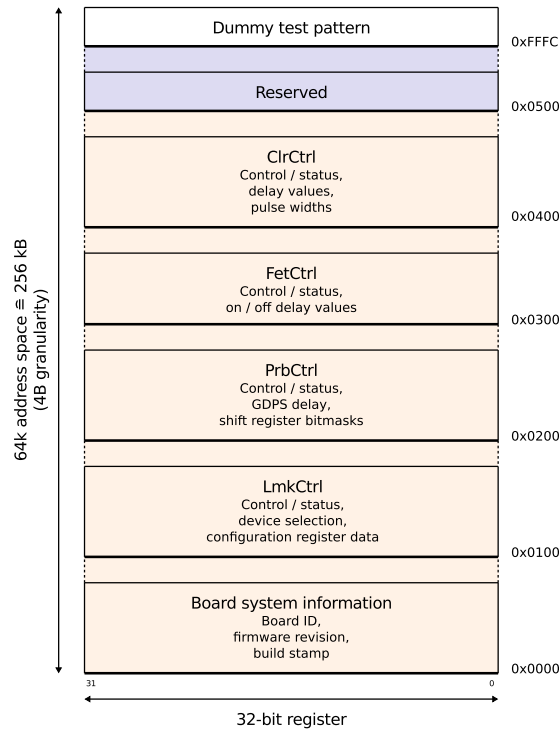
System configuration and control is realized on two levels. Low-level configuration of the FPGA device is enabled via the JTAG interface and is used for upgrading the firmware or re-configuration of the fabric. High-level configuration of the detector system is realized using a remote control mechanism, which enables access to system-internal configuration registers.

#### SysConfig Register Bank

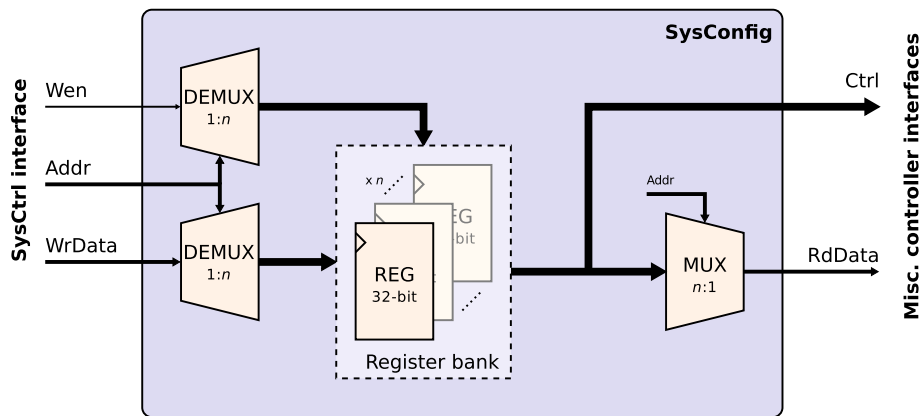
The SysConfig comprises both configuration data for the peripheral electronics and control registers to operate the controller modules. Full access to the SysConfig data is exclusively provided to the PPT through a dedicated controller module, while the peripheral controllers are granted read access only. The general structure and organization of the register bank is depicted in figure 7.4. The register bank provides a 16-bit wide address space (i.e. 65,536 registers max.) equally divided into segments of 256 (0x100) registers. As a register is of 32 bits in width, a 4-byte granularity is provided by the addressing mechanism, yielding a total memory capacity of 256 kB.

The first register segment (0x0000 to 0x00FC) is reserved for board-specific system information such as a unique board ID as well as revision and build stamp of the firmware. The segment between 0x0100 and 0x01FC buffers data for configuration and control of the Main Board fanout clock buffers through the LmkCtrl entity. Control and status information of the PrbCtrl module are provided by the segment from 0x0200 to 0x02FC, which additionally stores the shift register bitmasks that define the state of the ASIC power nets. The FetCtrl module is operated through the data stored in the subsequent segment (0x0300 to 0x03FC). The registers basically include the delay values for trimming the activation sequence accordingly. The last segment used for peripheral configuration data (0x0400 to 0x04FC) is assigned to the ClrCtrl entity and features both control and status information of the module as well as the pulse widths of the clear pulses. The subsequent address space starting from 0x0500 is reserved for future purposes. A special dummy register is located at address 0xFFFFC, which can be used as test pattern source for various testing purposes. All memory regions for the peripheral controllers share a common feature. The first register (offset +0x00 is always the Control Status Register (CSR) that operates the module and captures module status data.

As illustrated in figure 7.5, the logic for SysConfig incorporates a number of  $n$  32-bit registers (with  $n \leq 64k - 1$ ) and both a Multiplexer (MUX) and two De-Multiplexers (DEMUXs). When writing



**Figure 7.4:** Simplified structure of the SysConfig register bank.



**Figure 7.5:** Schematic of the logic implemented by the SysConfig register bank. The address serves as selector for both the MUX and the DEMUXs in order to specify the register to read from or write to, respectively.

data, the DEMUXs serve as address decoders and provide both the data to be written and the write enable flag to the corresponding register. On read access, the address acts as input selector to the MUX, which passes the register data of the selected register to the SysConfig output. The signals which operate the controller modules are provided directly via dedicated interfaces towards the entities.

Detailed information about the present SysConfig layout is given by the individual register maps

presented in chapter B.

## System Controller

The SysCtrl is the central part of the IOB slow-control sub-system that enables the PPT to both read and write access the IOB SysConfig. The controller acts as a slave device to the PPT, that is, it remains passive until the master entity transmits a serial data stream via the bi-directional data line CNTR. The master also provides a (differential) transmission clock over the SYNC signal line for synchronization.

**Serial Communication Protocol** Communication between the SysCtrl master (PPT) and the slave controller (IOB) is realized using a custom serial protocol. Its structure is described in table 7.1. When the master starts communication, the first two bits transmitted are always "10" and represent

Command	Start bits	Command bits	Payload	Description
READ	10	1000	<Addr> (16) + <RdData> (32)	Reads 32-bit data from address
WRITE	10	1001	<Addr> (16) + <WrData> (32)	Writes 32-bit data to address

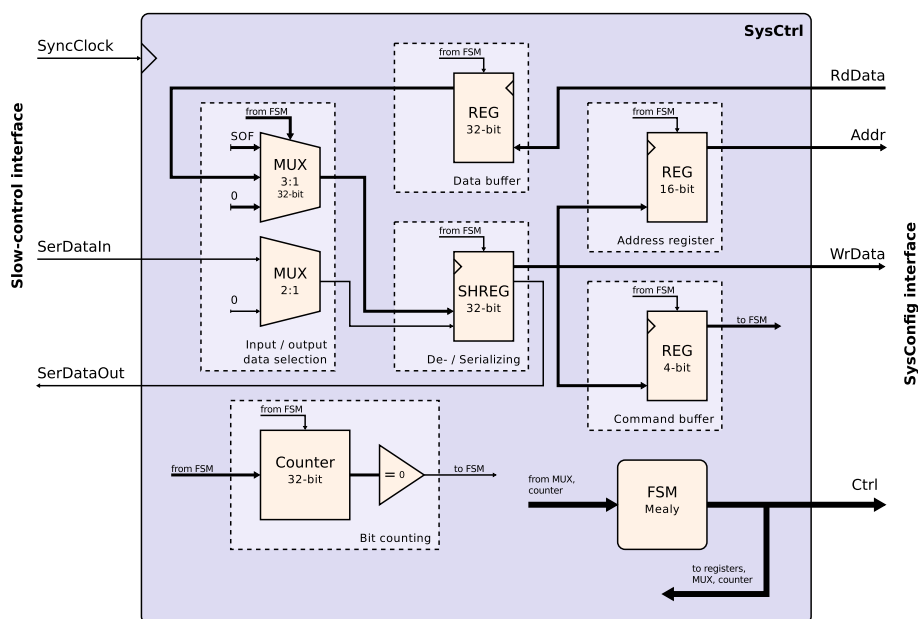
**Table 7.1:** Communication protocol of the slow-control interface between SysCtrl master and slave modules.

the Start of Frame (SOF) sequence. The subsequent four bits define the command, which are "1000" for read access and "1001" for write access to the SysConfig. In its initial draft, the protocol was foreseen to implement not only commands for read / write access to SysConfig, but also for advanced tasks like performing read / write bursts, dedicated control commands for the modules, and others. On the other hand, the extensive use of the basic implementation concluded that it is most convenient to realize more sophisticated tasks in software which is based on the SysConfig read / write commands. However, the number of four command bits has been preserved to provide flexibility for future implementations. The subsequent 16 bits specify the address of a SysConfig register. In case of a read access, the master waits for 32-bit data being transmitted from the slave controller. To signalize the beginning of the data transmission from the slave, data are preceded by the SOF sequence as well. In order to avoid blocking, the master should implement a timeout failsafe mechanism. When performing a write access, the 32-bit data are sent by the master immediately after the address bits. That is, the protocol currently lacks a handshaking mechanism that verifies the command has been accepted, which is advisable in future implementations.

A simplified illustration of the logic of the SysCtrl module is given in figure 7.6. On the slow-control interface side, the controller provides two lines for serial data input and output, respectively, and one for the transmission clock. As the data signal of the slow-control interface is bi-directional, both SysCtrl input and output data signals connect to a single FPGA I/O pad using a bi-directional FPGA I/O Buffer (IOBUF) primitive. The output-enable signal controls the direction of the IOBUF.

The interface towards the SysConfig provides two separate 32-bit data busses for read and write data, a 16-bit address line, and a few control signals for handshaking and write-access control.

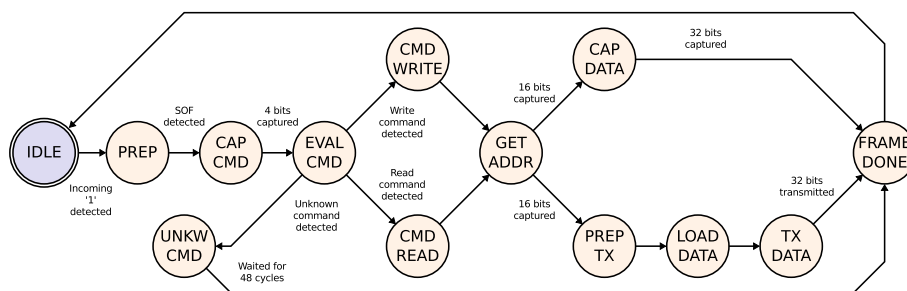




**Figure 7.6:** Simplified schematic of the SysCtrl logic concept. The data path comprises a 32-bit shift register for the incoming and outgoing data (both serial and parallel). Two separate registers buffer both the command, which defines whether a read or write access is requested, and the address that specifies the SysConfig register.

The internal logic of the SysCtrl data path comprises a 32-bit shift register as central element, which provides both serial and parallel input and outputs. The MUX located in front of the parallel shift register input selects the data source (i.e. SOF or data) when transmitting towards the master. Two registers at the parallel shift register output buffer the 4-bit command and the 16-bit address for accessing the SysConfig, respectively.

The main element of the control path is a Mealy-type FSM. It operates a decrementing counter, which clocks the number of bit shift cycles. Reaching zero counter triggers FSM state transitions and certain actions like raising flags for enabling the (shift) registers or loading a new counter start value. Figure 7.7 shows the transition graph of the SysCtrl FSM.



**Figure 7.7:** State transition graph of the SysCtrl FSM.

The FSM start state is represented by **IDLE**. An incoming '1' on the data input line triggers a transition to the **PREP** state, which checks whether the SOF sequence has been detected. If so, the

counter is prepared for the number of incoming command bits, and state **CAP\_CMD** is entered, which enables the shift register for command capturing. When the counter reaches zero, the FSM raises a flag to internally store the command and proceeds with command evaluation in state **EVAL\_CMD**. The present FSM implementation recognizes SysConfig read and write commands only. In case of an unknown command, state **UNKW\_CMD** is entered. This state waits for another 48 cycles as a safety measure. If the FSM would immediately return to **IDLE**, possible incoming address (16 bits) and data (32 bits) from the master could accidentally trigger another FSM run and lead to unpredictable behavior.

Depending on the outcome of command evaluation, the FSM enters **CMD\_READ** or **CMD\_WRITE**, which set a command evaluation flag accordingly. Both of the two states prepare the counter for the number of incoming address bits and proceed with state **GET\_ADDR** to capture them. When the counter reaches zero, the bits are stored into the internal address register. At this time, the FSM decides whether to proceed with the read or write branch, depending on the command evaluation flag.

In case of a read command, the FSM enters state **PREP\_TX**, which asserts the output enable signal of the IOBUF primitive for data output as required to transmit the SOF sequence. The following state **LOAD\_DATA** loads the SysConfig data stored in an internal register, prepares the counter with the number of data bits, and proceeds with transmission in state **TX\_DATA**.

When a write command has been detected, state **CAP\_DATA** captures the next 32 incoming bits on the serial data line. With the counter reaching, the write-enable signal is asserted to store the data from the shift register into the register specified by the internally buffered address.

Both read and write transactions are completed by state **FRAME\_DONE**. It acts as a finalizing "clean-up" state that brings the SysCtrl into a defined state by clearing the shift register and resetting the FSM enable flag.

### 7.1.3 Peripheral Control

Peripheral electronics is operated by several dedicated controller modules, which are presented in the following. The current implementation allows manual module control through SysConfig. The final firmware concept, however, foresees autonomic operation administrated by the IOB master FSM while preserving manual control for testing purposes.

## Control of Analog and Digital ASIC Power

Switching both the analog and the digital ASIC supply voltages is managed by two modules, the PrbCtrl and the PrbSRCtrl, which operate the PRBs of a sensor Main Board.

A PRB controls the nets of each power net group of four ASIC devices. In particular, two daisy-chained 8-bit shift register residing on a PRB are programmed with a bitmask that defines the state of the individual power nets. As the four PRBs are daisy-chained as well, a total number of 64 bits must be transmitted into the shift registers. Table 7.2 illustrates the relation between the shift register outputs, the power nets, and the position of a bit within the data stream. The first two lines explain the relation between the shift register outputs and the power nets, in which the letters 'A' to 'D' denote the four ASICs, followed by the number of the power net group<sup>4</sup>. The numbers of the section below show the bit assignment of the data stream. The bits are shifted in MSBit first, that is bits 63 to 48 control the power nets of PRB #4.

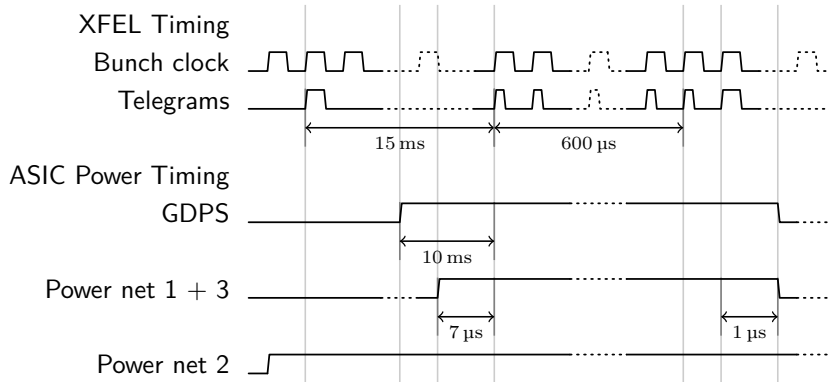
---

<sup>4</sup>In the following, *(power) net x* refers not only to a single net, but also to the group number

SR output	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Power Net	n/c	n/c	n/c	GDPS	D3	D2	D1	C3	C2	C1	B3	B2	B1	A3	A2	A1
PRB	#1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	#2	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	#3	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
	#4	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62

**Table 7.2:** Bit assignment of the PRB registers. The first two rows describe the relation between the 16-bit shift register and its assigned power net control line. The section below explains the bit assignment in the 64-bit data stream, which is transmitted MSBit first. The number represents the bit index, i.e. bit 63 controls power net A1 of PRB #4.

The purpose of PrbCtrl is implementing a programming sequence that controls the power nets according to the timing given in figure 7.8. The module fetches configuration data from SysConfig



**Figure 7.8:** Preliminary timing diagram for the ASIC power nets.

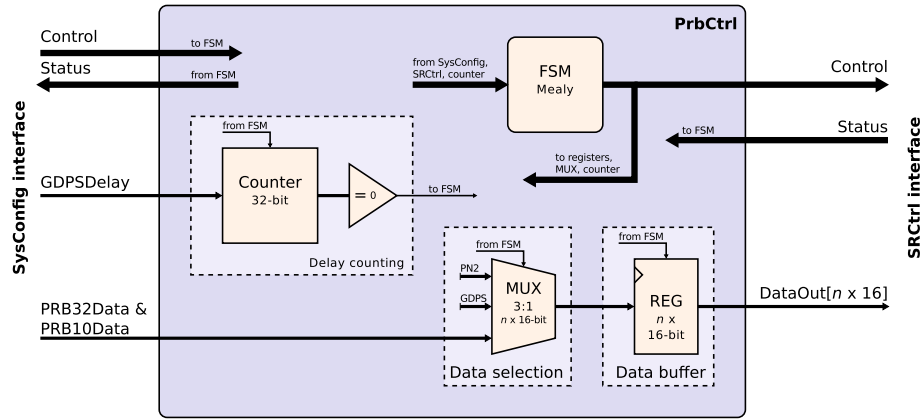
and provides them to PrbSRCtrl, which performs the actual shift register programming by operating on the control signals connected to the FPGA I/O interface.

The ASIC supply voltages are grouped into three types of power nets. Group 1 represents the power for digital ASIC electronics that is required to be active during the macro-bunch period only and thus is switched. The power nets in group 2 also supply digital electronics, but must be permanently enabled during the entire period of the experiment. Power for analog ASIC electronics is provided by the nets of group 3, which are also turned off during the inter-train gap. In order to provide sufficient margin to settle and stabilize, the supply voltages of group 1 and 3 must be switched on about 7  $\mu$ s before the first bunch arrives. After the XFEL stop telegram is received, all three power nets should kept active for another 1  $\mu$ s to avoid negative effects resulting from overshoots or undershoots, for example. A special power net is GDPS, the power supply for the sensor power gate drivers. The drivers require at least 10 ms of settling and stabilizing margin. Similar to the ASIC power supply, the gate drivers should be switched off about 1  $\mu$ s after the XFEL stop command.

The global timing for shift register configuration will be provided by the IOB master FSM synchronously to the XFEL electron bunch timing. That is, PrbCtrl will be enabled every 100 ms (10 Hz train repetition rate) with a duration of about 608  $\mu$ s to cover the active part of a train period. As the bitmasks stored in the PRB shift registers directly reflect the state of the power nets, they need to be reprogrammed three times within a 10 Hz cycle. The first programming cycle enables GDPS. Power nets 1 and 3 are enabled by the second configuration cycle. The last cycle

disables all supply voltages but power net 2. The PrbCtrl implements the required local timing for the programming sequence by a delay mechanism and an FSM which triggers on both rising and falling edges of the controller enable signal.

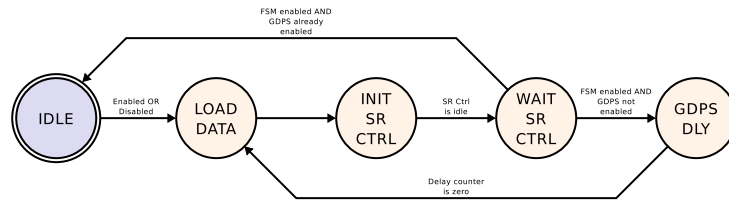
Figure 7.9 shows the schematic of the PrbCtrl module. The bitmasks from SysConfig are provided



**Figure 7.9:** Schematic of the PrbCtrl module, which serves as master controlling entity to the PrbSRCtrl.

to a MUX, which selects between three different data sources according to the local programming cycle number. The selected data are buffered in an output register for PrbSRCtrl. A decrementing counter serves as timer for the delay mechanism and retrieves its start value from another register of SysConfig.

A Mealy-type FSM implements the sequencing of the programming cycles and also generates the control signals for the PrbSRCtrl entity. The FSM transition graph is illustrated in figure 7.10. Enabling the controller triggers a transition from **IDLE** to **LOAD\_DATA**. This state produces the

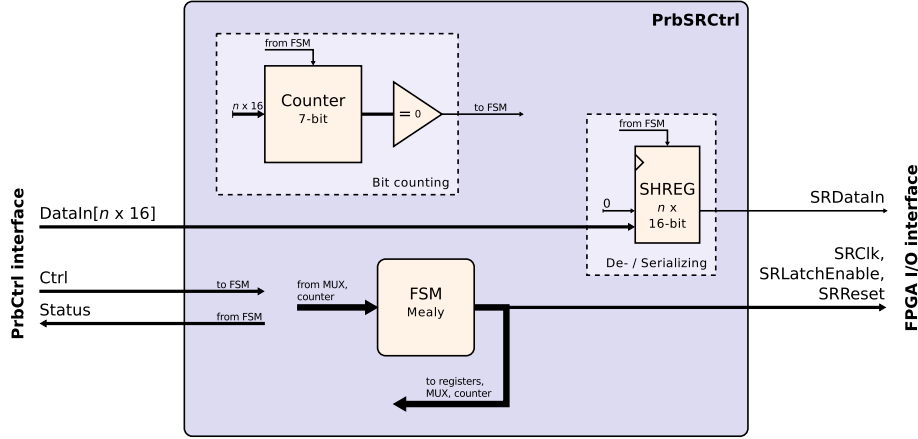


**Figure 7.10:** Transition graph of the FSM for the PrbCtrl module.

selection signal for the data MUX depending on the status of controller and counter. It also latches the data into the output buffer, and subsequently proceeds with **INIT\_SR\_CTRL**. In case PrbSRCtrl is idle, the FSM asserts the required control signals which initiate the shift register programming and waits for acknowledge in state **WAIT\_SR\_CTRL**. Depending on whether power net GDPS has already been activated or not, the FSM enters state **IDLE** or **GDPS\_DLY**, respectively. Proceeding with **GDPS\_DLY** enables the counter to decrement until a zero flag is raised, which stops counting and triggers a transition back to **LOAD\_DATA** that starts the programming cycle with SysConfig data. This time, the still asserted zero flag causes the FSM to return to **IDLE** when **WAIT\_SR\_CTRL** is entered. De-asserting the controller enable signal causes another transition into **LOAD\_DATA**. However, according to the controller status, the MUX selects the configuration pattern for disabling

all nets but power net 2. The programming cycle is initiated, and the FSM returns to **IDLE**. In any case, a dedicated abort signal can interrupt the programming sequence and immediately returns the FSM to **IDLE**. The abort signal is also provided to PrbSRCtrl.

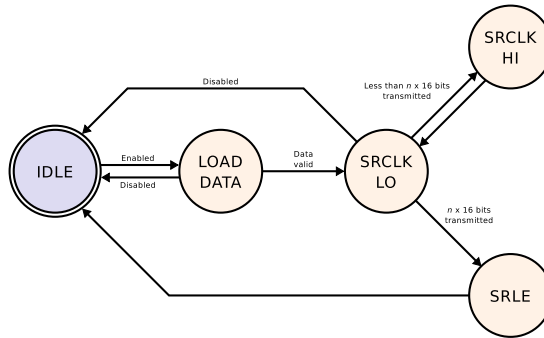
The schematic of the PrbSRCtrl module is presented in figure 7.11. The input data provided by



**Figure 7.11:** Schematic of the PrbSRCtrl module, which provides the access to the PRB shift registers.

PrbCtrl is loaded into an internal shift register of corresponding width, and a decrementing counter clocks the number of bits being pushed into the PRB shift registers. The serial output of the internal shift register connects to the input line of the PRB shift register interface.

The Mealy-type FSM as central element implements the programming mechanism and provides the required control signals towards the FPGA I/O interface. Its state transition graph is illustrated in figure 7.12. Starting from idle state **IDLE**, enabling the controller triggers a transition into



**Figure 7.12:** Transition graph of the FSM for the PrbSRCtrl module.

**LOAD\_DATA.** When the data provided on the input interface are marked as valid, they are loaded into the internal shift register, and the bit counter is initialized with a corresponding start value. The FSM proceeds with **SRCLK\_LO**, which generates the low-state of clock signal for the PRB shift register and decrements the counter by one. In case there are bits left for transmission, the FSM proceeds with **SRCLK\_HI**. This state generates the high-state of the clock signal. Additionally, the internal data are shifted by one, which assigns the next bit to the output line. Subsequently,

**SRCLK\_LO** is re-entered. If all data bits have been transmitted (i.e. the bit counter is zero), the FSM proceeds with **SR\_LE**, which asserts the latch enable signal to store the bitmask and returns to **IDLE**.

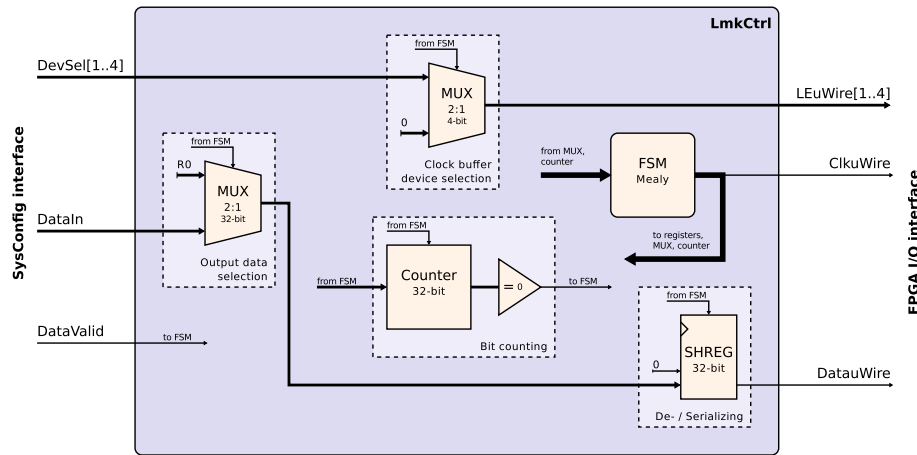
Both PrbCtrl and PrbSRCtrl provide a generic VHDL code implementation that allows them being adjusted to the number of PRBs present in a particular test setup. Basically, the data width of several logic components like the MUXs, the internal (shift) registers, and the corresponding signal busses, is calculated depending on the number of boards defined in the VHDL code.

## Mainboard Fanout Clock Buffer Control

The module LmkCtrl implements the programming mechanism for configuring the fanout clock buffers mounted on the sensor Main Board, which distribute a copy of the ADC sampling clock and the XFEL FEE clock to each ASIC. Internal configuration registers allow adjusting the input-to-output signal delay<sup>5</sup> individually for each clock output of the fanout devices. As the distributed clock signals are required to have a well-defined phase relation with regard to the electron bunch timing, the delay fine adjustment is an essential feature to compensate for clock skew induced by signal processing and differences between the lengths of the individual signal paths.

Configuration of the Main Board fanout clock buffers is part of the detector calibration phase, which is triggered and managed by expert control software. The programming sequence does not require for bunch-synchronous timing, but will be initiated at the beginning of an experiment run. That is, LmkCtrl does not need to have permanent access to configuration data for the individual registers of each clock buffers. Further, the registers of a device are programmed independently from each other. The devices expect a serial transmission of 32-bit data words into an internal shift register. A latch-enable signal completes the transmission cycle and stores the shift register content accordingly. Therefore, it is reasonable to let the software provide the individual data words to LmkCtrl instead of storing them all in SysConfig registers.

The schematic of the LmkCtrl module is presented in figure 7.13. The data path incorporates a

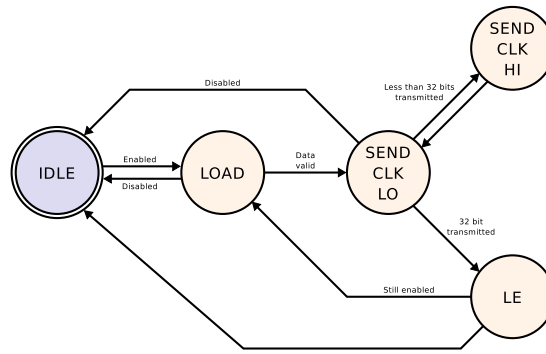


**Figure 7.13:** Schematic of LmkCtrl. The controller comprises two MUXs which manage the output data selection and the device selection. A shift register converts the parallel configuration data word into a serial bit stream.

<sup>5</sup>From 0 ps to 2250 ps in steps of 150 ps, cf. [55]

MUX for output data selection and a 32-bit shift register, which transmits the data bits towards the clock buffers. Configuration data are provided to the MUX by a single SysConfig register, which is updated with every register programming cycle. The serial output data line of the internal shift register connects to the clock buffers in parallel, that is, each device is provided with the same data stream. Asserting the individual latch enable signals eventually assigns the data to the corresponding device(s). Thereby, the controller enables simultaneous programming of multiple devices. A MUX for device selection controls the behavior of the latch enable signal according to the timing specification given in the clock buffer data sheet [55]. It ensures the latch enable signals to be de-asserted during bit transmission according to the device selection mask supplied by the SysConfig, and re-asserting them when finished.

Figure 7.14 describes the transitions of the Mealy-type FSM, which is the central element of the control path and steers both the transmission clock and latch enable signals accordingly. Starting



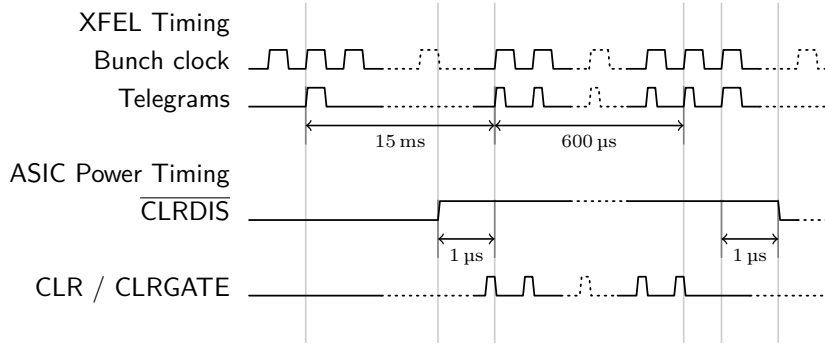
**Figure 7.14:** State transition graph of the FSM for the LmkCtrl module.

from **IDLE**, enabling the controller causes a transition into **LOAD\_DATA**, which waits for data from SysConfig. If valid, data are stored into the internal shift register. Additionally, a decrementing counter is initialized with the number of bits to be transmitted. Next, the FSM proceeds with **LMKCLK\_LO** which generates the low-signal part of the serial data transmission clock. In case there are bits left for transmission, state **LMKCLK\_HI** is entered, which asserts the transmission clock, and also shifts the data bits by one, before it re-enters the previous state. If all bits are transmitted (i.e. the counter has reached zero), a transition to **LMK\_LE** is triggered that re-asserts the latch enable signals. Depending on the controller status, the FSM returns to **IDLE**, or waits for another valid data word.

As an LMK01010 clock buffer incorporates 10 registers, and a sensor Main Board houses four of those devices, the programming cycle has to be executed 40 times, corresponding to a maximum data volume of 1280 bits per sensor module. Considering a transmission clock period of 50 ns (20 MHz), configuration takes about 64  $\mu$ s plus latency induced by control software.

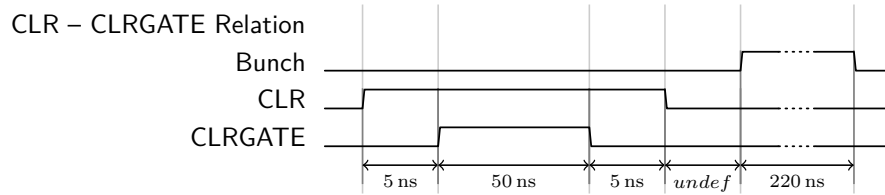
## Sensor Clear Signals

The mechanism to control the sensor clear signals is provided by the ClrCtrl module. The timing of the control sequence is required to be phase-synchronous to the XFEL electron bunch timing, as the preliminary draft in figure 7.15 illustrates. The low-active signal  $\overline{\text{CLRDIS}}$  disables the gate drivers located on the PRBs. It must be de-asserted about 1  $\mu$ s before the first bunch arrives and remain inactive until 1  $\mu$ s after the XFEL stop telegram. Regarding **CLR** and **CLRGATE**, the two signals have



**Figure 7.15:** Preliminary timing of sensor clear signals.

to be applied to the sensors before each bunch to clear both drain and gate contacts. The time distance between finishing the clear cycle and the incoming bunch is yet undefined and has to be determined experimentally. As shown in figure 7.16, the pulse of **CLRGATE** remains within the pulse of **CLR**. For the rising edges, a **CLR**-to-**CLRGATE** delay of 5 ns is aimed for. The **CLRGATE**-to-**CLR** delay



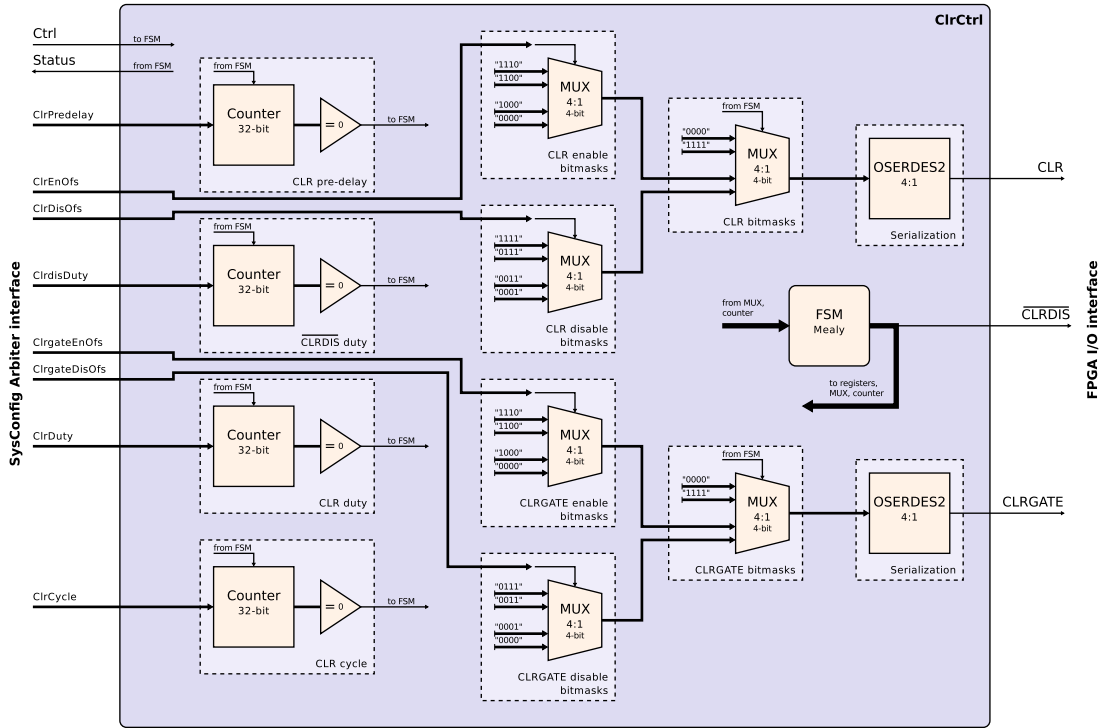
**Figure 7.16:** Preliminary timing of **CLR** and **CLRGATE** signals. The clear gate pulse lies within the time window of a clear (drain) pulse. The time distance between finishing the clear cycle and the incoming bunch is yet to be determined experimentally.

on the falling edges is in the same order. The flat-top duration of **CLRGATE** changes accordingly depending on the flat-top period of **CLR**, which is initially foreseen to be in the order of 60 ns.

As for the timing, the IOB master FSM triggers the controller state machine at the train repetition rate of 10 Hz and an offset relative to the XFEL start telegram. Internal logic generates the pulse sequences for the three clear signals to obtain the local bunch-related timing.

Figure 7.17 shows a diagram of the **ClrCtrl** logic. Four decrementing counters serve as timers for producing the pulse durations of **CLRDIS**, **CLR**, and **CLRGATE** signals. The counter start values are defined via **SysConfig** registers for conveniently adjusting them during sensor calibration. The pulse width of **CLRDIS** is defined by the value provided via **CLRDISDUTY**, while the delay between the rising edges of **CLRDIS** and the first **CLR** pulse is specified by **CLRPREDelay**. The special relation between the **CLR** and **CLRGATE** pulses is implemented using two Output Serializer/Deserializer (**OSERDES**) primitives of the FPGA, which allow parallel-to-serial data conversion at different SerDes ratios. The **OSERDES** of the **ClrCtrl** uses a SerDes ratio of 4:1. That is, during each input clock cycle, a four-bit wide input data vector is provided and serially clocked out within the same period. Consequently, the **OSERDES** must additionally be provided with an output clock of fourfold input frequency. By applying selected input bitmasks to the **OSERDES**s of the clear and the clear gate signal, the edges of the output signal can thus be shifted by an offset which corresponds to a multiple (0..3 or 1..4) of the output clock period. Considering the 99 MHz XFEL FEE clock as input reference, the output clock is of 396 MHz yielding a shifting granularity of about 2.5 ns,



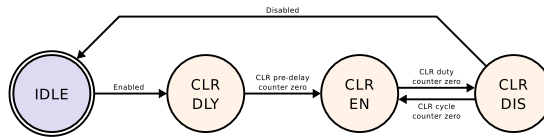


**Figure 7.17:** Schematic of the ClrCtrl module. Four counters generate the signal delay values to generate the clear signal timing. A set of MUXs is used to select the appropriate edge-to-edge offset values for both clear and clear gate control signal. The OSERDES instances eventually connect the controller the FPGA I/O interface.

which is sufficient regarding the requirements of the sensor.

The shifting offsets are specified by SysConfig registers and delivered via `CLRENOFS` / `CLRDISOFS` and `CLRGATEENOFs` / `CLRGATEDISOFS`, while the mechanism for selecting the proper input bitmask is done via cascaded MUXs for each of the two clear signals.

The Mealy-type FSM steers internal control signals with regard to the counter states. Its transition graph is shown in figure 7.18. Starting from **IDLE**, enabling the controller results in entering



**Figure 7.18:** State transition graph of the FSM for the ClrCtrl module.

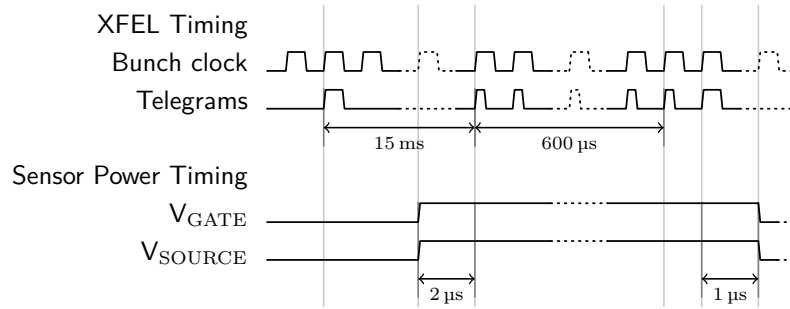
state **CLR\_DLY**. This state de-asserts `CLRDIS` and also enables the pre-delay counter, which defines the time between the rising edges of `CLRDIS` and `CLR`. When the counter reaches zero, the FSM enters **EN\_CLR**, which is responsible for asserting `CLR` for the number of cycles specified by the start value of the clear-duty counter. The FSM waits until the number of cycles have passed, and subsequently proceeds with **CLR\_DIS**. In this state, the clear signal becomes de-asserted for as long as the clear-cycle counter is not zero. The clear-cycle counter defines the period of a full clear cycle (i.e. high-low) and is enabled simultaneously with the clear-duty counter. Thereby, the clear pulses

can be adjusted to different electron bunch patterns. When the counter has reached zero, the FSM re-enters **CLR\_EN** and triggers another clear cycle, until the controller is disabled and **IDLE** state is returned to. A fourth counter is used to define the duty cycle of signal  $\overline{\text{CLRDIS}}$  and is triggered with enabling the controller.

## Sensor Power Switching

The FetCtrl entity manages switching the supply voltages of the DEPFET sensors, which are active during the macro-bunch period and turned off during the inter-train gap. For that purpose, the module operates the control signals of onboard FET drivers connected to the FETs that actually gate the sensor power nets.

In order to allow the supply voltages to stabilize, the sensor power requires being switched early with regard to the first incoming bunch. Similarly, after the last arriving bunch, the sensors should be kept powered for a specific time. Figure 7.19 explains the preliminary timing of the new sensor power switching scheme (omitting  $V_{SSS}$  as active power net) with regard to the XFEL timing. An offset of at least  $2\mu\text{s}$  before the first incoming bunch is recommended for both  $V_{\text{GATE}}$  and



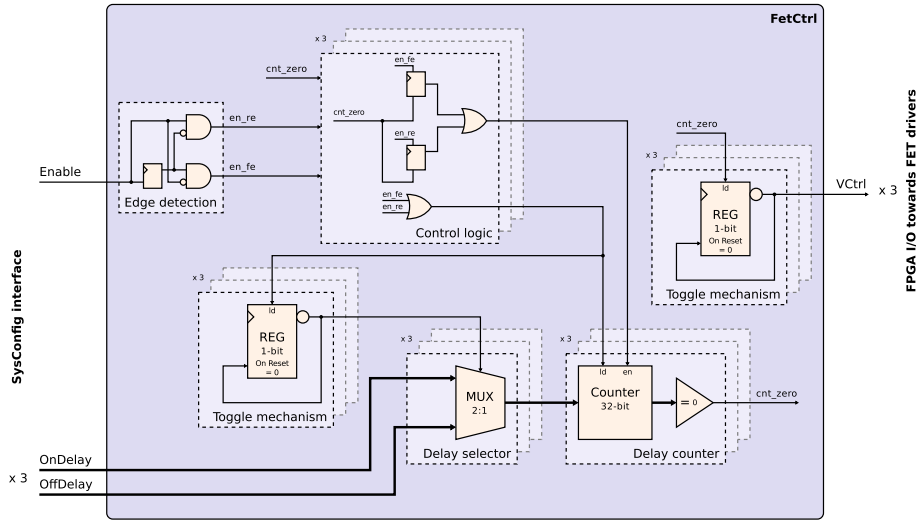
**Figure 7.19:** Preliminary timing of the new sensor power switching scheme without  $V_{SSS}$ . Both  $V_{\text{GATE}}$  and  $V_{\text{SOURCE}}$  should be turned on at least  $2\mu\text{s}$  before the first bunch arrives in order to provide sufficient time margin for the power to stabilize. After the XFEL stop telegram, power should remain active for  $1\mu\text{s}$ .

$V_{\text{SOURCE}}$ , while the power should be kept active for another  $1\mu\text{s}$  after the XFEL stop telegram has been received.

The IOB master FSM provides the global controller timing, which enables the module at 10 Hz train repetition rate, phase-synchronous to the electron bunch timing. As the power nets have to be active during the entire period of incoming bunches, the supply nets will be enabled for about  $600\mu\text{s}$  including time margin for stabilizing and settling of the supply voltages. The FetCtrl implements a delay mechanism, which allows individually shifting the sensor power control signals relatively to the controller enable signal. In that way, both the order and the delay offsets can be fine-adjusted to determine the settings for best detector performance.

The logic incorporated by the FetCtrl entity is outlined in figure 7.20. In contrast to the other modules, its control path does not comprise a dedicated FSM, but is based on a few combinational logic elements, registers, and counters instead.

Basically, toggling the enable signal translates into toggling the FET driver control signals. As the switching order of the power nets is not yet conclusively defined, the controller additionally



**Figure 7.20:** Block diagram of the FetCtrl logic. Unlike the other modules, this controller does not feature a dedicated FSM as central element of the control path, but uses

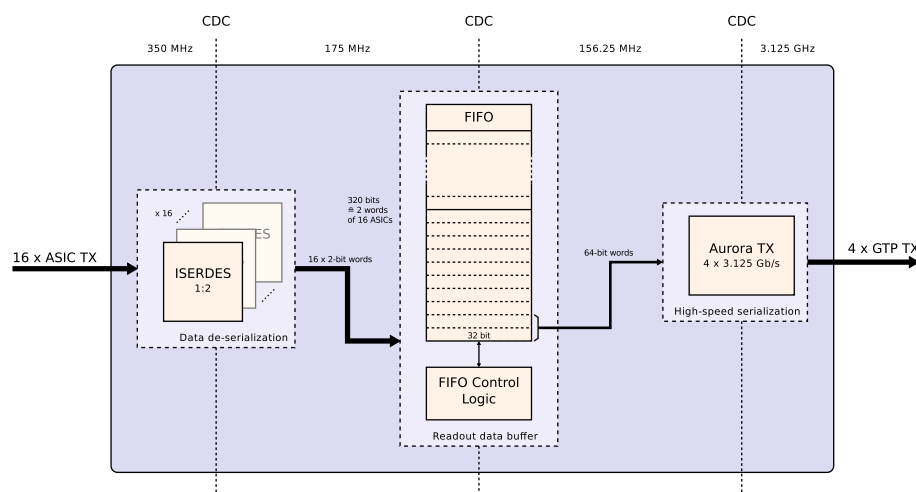
implements a mechanism which allows delaying the FET driver control signals relatively to both the rising and falling edge of the enable signal. The corresponding delay values (in multiples of controller clock cycles) are supplied via two data busses per sensor voltage stored in SysConfig.

An edge detection logic processes the enable signal and generates pulses for the internal control logic. Two registers implement the toggle mechanism for both the delay selector MUX and the outgoing FET driver control signals. On reset, the registers are initialized with a value of zero. The inverted register output is fed back to the register input. When the register load signal is asserted, the assigned input value is stored, and thus the output toggles. The control logic uses both the edge detection pulses and the counter zero flag as input and generates the register load strobes as well as the counter load and enable signals. In order to select between different delay values for the enabling and disabling sequence, the strobe of the delay selector toggle mechanism pulses on each rising and falling edge of the controller enable line. As a consequence, the MUX select signal toggles likewise with each generated pulse. On the other hand, the FET driver control signal inverts its state with the counter reaching zero.

A copy of the control logic and the toggling registers as well as of the delay selector and the delay counter exists for each switchable sensor voltage. As the IOB prototype is designed for switching  $V_{GATE}$ ,  $V_{SOURCE}$ , and also  $V_{SSS}$ , the current implementation of the FetCtrl incorporates three duplicates of gating logic.

#### 7.1.4 ASIC Data Transport

Figure 7.21 describes the concept for the data transport during ASIC readout. Frame data are transmitted by the ASICs at a frequency of 350 MHz single data rate and captured in a deserialization stage of the IOB FPGA. The Input Serializer/Deserializer (ISERDES) primitives convert the serially incoming bit stream of each input channel into two-bit wide data chunks, which are packetized to a 32-bit word and buffered in a FIFO. As the current SerDes ratio is 1:2, the FIFO input clock domain is of 175 MHz. The FIFO output provides 64-bit wide data words in



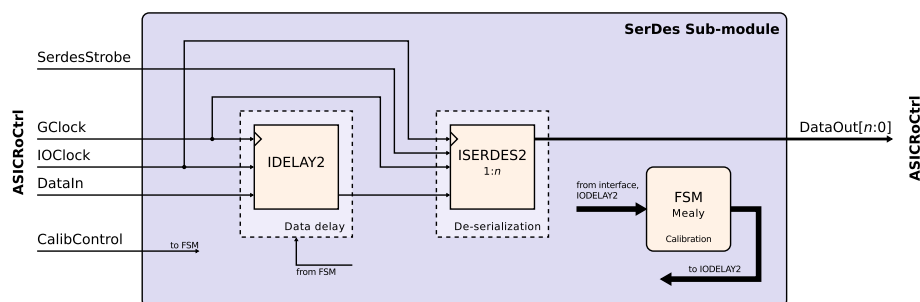
**Figure 7.21:** Concept of ASIC readout.

a 156.25 MHz clock domain to the Xilinx Aurora core, which converts them into four 3.125 Gb/s high-speed serial bit streams that are transmitted via the GTPs using the Aurora protocol.

The initial draft of the high-speed data transmission, which is presented in the following, uses all four GTP lanes. However, since the number of transceiver lanes of the PPT has been reduced to three per IOB, the readout mechanism has to be modified accordingly.

## Data De-serialization and Buffering

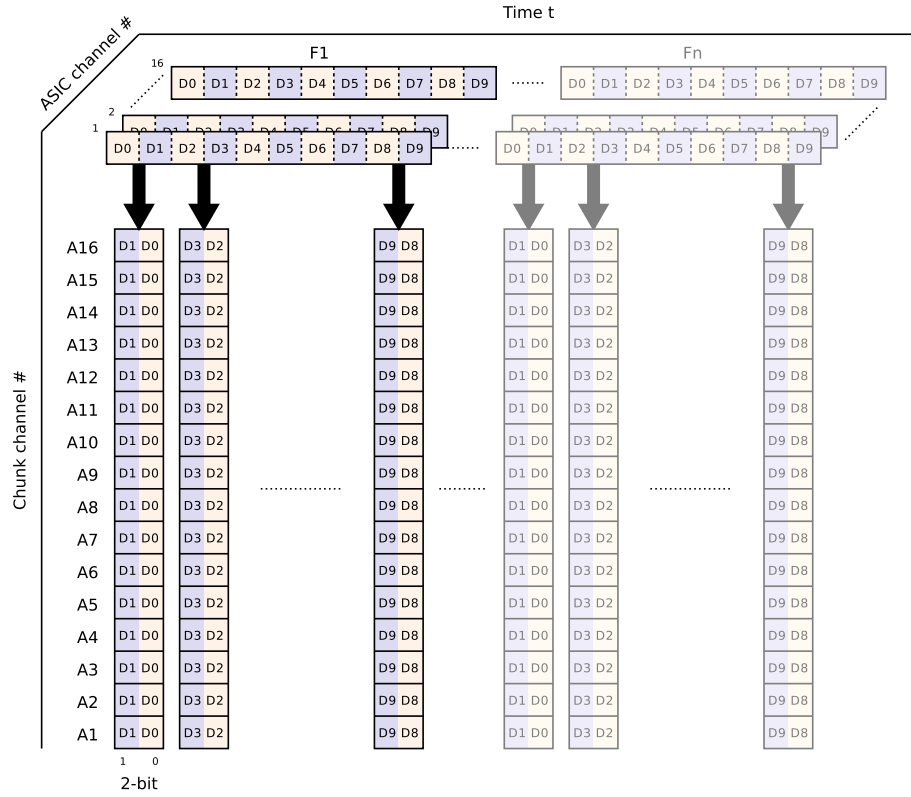
Both de-serialization and buffering of the serial ASIC data are handled by the AsicRoCtrl. The logic for the SerDes mechanism is implemented in a sub-module, of which one is instantiated for each data channel. As the schematic in figure 7.22 illustrates, the sub-module data path basically comprises an IODELAY and an ISERDES. The IODELAY allows shifting the data with regard



**Figure 7.22:** Schematic of the de-serialization module using ISERDES primitives.

to the transmission clock in order to align the edges for proper data capturing. In the present implementation, the amount of delay is specified manually in the VHDL code and needs to be adjusted during detector calibration. The delayed bit stream is provided to the ISERDES, which de-serializes the data according to the SerDes ratio 1:n and builds  $n$ -bit words. A Mealy-type FSM controls the calibration mechanism required for proper IODELAY setup.

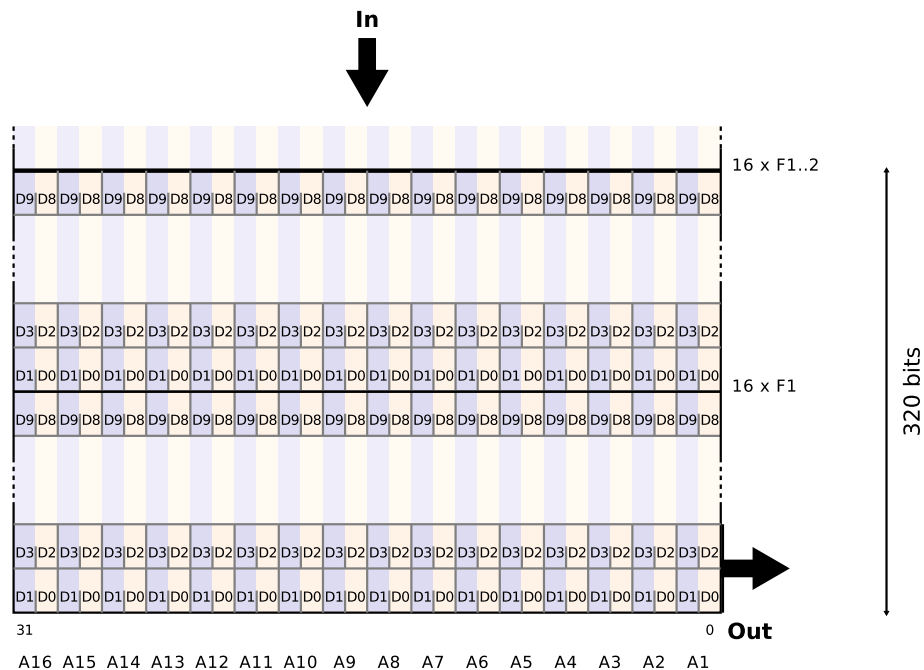
The draft layout of the packetized 32-bit data words is shown in figure 7.23. The incoming bits of



**Figure 7.23:** Serialization of ASIC data words. Two subsequent bits build a frame chunk. The chunks of all ASICs are combined to a 32-bit data word.

an ASIC frame data word  $F1$  are ordered LSBit first, that is, data bit  $D0$  is put on bit 0 within a frame chunk. For each data channel  $n$ , a corresponding chunk channel  $A_n$  exists, represented by the output of an ISERDES sub-module. The chunks are combined to a 32-bit data word  $W$  in ascending order, that is, chunk channel  $A0$  is always put into the first  $n$  bits of  $W$ . Accordingly, the content of the FIFO is sorted as shown in figure 7.24. As each data word  $W$  contains two bits of ASIC data word  $F_n$ , a complete frame data word for all ASIC channels is built from five consecutive FIFO words, corresponding to 80 bits. The FIFO outputs 64-bit wide data (two consecutive FIFO words) that match the Aurora input data width. As a consequence, an offset occurs between the boundaries of frame data and FIFO output data. As the boundaries superpose every 320 bits, data are transmitted on a split-frame basis.

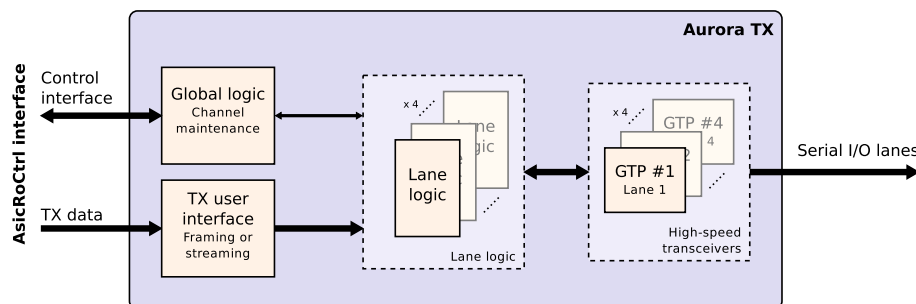
The code of AsicRoCtrl module provides a generic implementation regarding the number of serial data input channels and SerDes ratio, respectively. By specifying both constants in the VHDL code, the module can be adapted for operation under different test conditions. The present implementation supports SerDes ratios of up to 1:4. However, a ratio of 1:2 is aimed for and should be sufficient. Applying higher SerDes ratios also includes a change in the offset of the word-boundaries.



**Figure 7.24:** Layout of ASIC readout FIFO. The 32-bit words contain two subsequent data bits of all 16 ASIC data words. Consequently, a complete ASIC data word for each chip is built from five consecutive FIFO words.

## High-speed Data Transmission with Aurora

High-speed data transmission towards the PPT is realized with the Xilinx Aurora core in TX-only configuration. A block diagram of the Aurora core is given in figure 7.25. The 64-bit wide TX data



**Figure 7.25:** Block schematic of Aurora TX core. Source: [56]

input bus connects to the FIFO output of AsicRoCtrl and provides the data to the transceiver lanes through the TX user interface. Initialization of the transceivers and encoding / decoding of control characters as well as error handling is managed by individual lane logic modules individually assigned to a GTP transceiver. The global logic of the Aurora core is responsible for channel bonding and verification during the channel initialization. It connects to the control interface of AsicRoCtrl and regulates the data flow between the two modules accordingly by indicating whether the core is ready for new data.

## 7.2 Simulation and Verification

In order to verify the functional behavior of the design, functional code simulations have been performed on both the custom-developed VHDL modules and the proprietary logic cores from Xilinx. As VHDL simulator tools, both proprietary *ModelSim*<sup>6</sup> and open-source *GHDL*<sup>7</sup> have been used. The result of a simulation is typically displayed as waveforms.

Simulating the design code before testing on the real FPGA has a number of advantages. The simulator allows monitoring internal signals, which are not visible on the physical boundary of the FPGA device. Also, simulation can be performed when the target hardware is yet unavailable. Functional simulation, on the other hand, does not properly consider precise timing behavior of the design. As a consequence, the results of functional simulation and the real design may differ. However, timing-accurate simulation is only feasible for very specific cases, as the execution time usually is higher by one or two orders of magnitude compared to that of functional simulations.

Verification of the design functionality on the real FPGA device is initially done by measuring signals on the I/O pins with an oscilloscope or logic analyzer. However, this limits the access to external signals only, which is not sufficient for complex FPGA designs. For also monitoring internal signals, samples from a selected set of signals are taken during runtime and stored in internal FPGA memory. The recorded data are then read out via JTAG and displayed in waveform view similar to that of simulations. The *ChipScope* analyzer tool provided by the Xilinx software environment uses this approach for design analysis.

In contrast to a functional simulation, the FPGA test method allows gaining valuable information about signal behavior during runtime under real-life conditions. However, the number of samples being recorded is usually limited to a few thousands, and sampling is done for a particular set of signals only. Observing other signals requires a modification of the signal set and also the time-consuming physical re-compilation of the design code. On the other hand, signal data of a simulation is stored for every (internal) signal at each moment of simulation (according to its time resolution).

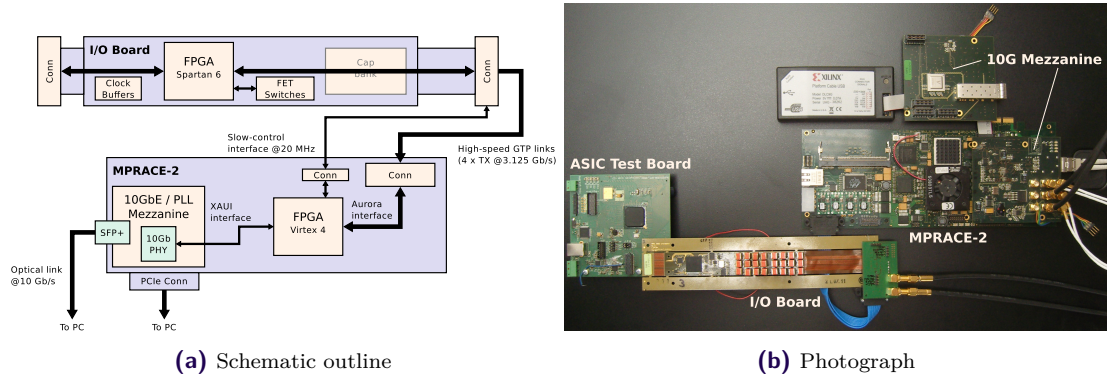
The results of functional simulation of the firmware modules are presented in chapter B., while the ChipScope waveforms are presented in the next chapter.

## 7.3 Test Environment

For verifying both the electrical operation of the IOB prototype and the functionality of its FPGA firmware, an initial prototype test environment has been set up. The test system comprises the IOB prototype and the MPRACE-2, a custom FPGA board developed at ZITI of Heidelberg University. The MPRACE-2 additionally carries a custom mezzanine which provides 10GbE capability and a PLL device. An additional adaptor board (not shown in the outline) realizes the interconnection between the two boards. Both mezzanine and adaptor board have been developed and designed within the scope of this thesis. Figure 7.26 (a) and (b) illustrate a schematic outline of the test environment and a photograph of the real components, respectively.

<sup>6</sup><http://www.model.com/>

<sup>7</sup><http://ghdl.free.fr/>



**Figure 7.26:** Block diagram and photograph of the IOB test setup.

### 7.3.1 The MPRACE-2 Board

The Massive Parallel Readout Accelerator v.2 (MPRACE-2) is a FPGA co-processor board with PCIe form-factor. It is based on a dual-FPGA approach using Xilinx Virtex-4 devices and has several general purpose features [40].

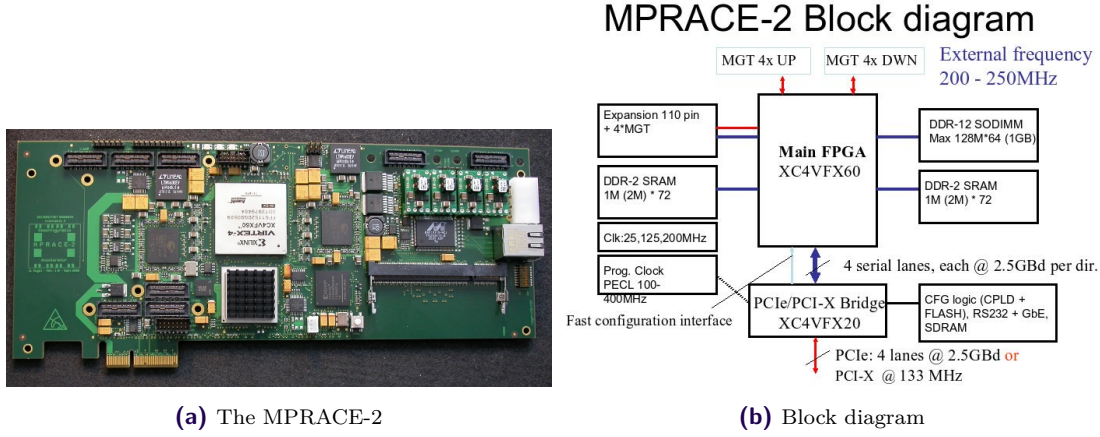
- A small bridge FPGA realizes a four-lane PCIe interface, while a large main FPGA implements the main functionality.
- Eight serial high-speed transceiver are used for communication between bridge and main FPGAs (4 x 5 Gb/s max. per lane) and board-to-board interconnection (4 x 2.5 Gb/s max. per lane). Another four high-speed links with up to 5 Gb/s per lane allow interfacing external devices via mezzanine connectors.
- A 1GbE interface connected to the bridge FPGA allows communicating with the MPRACE-2 over standard Ethernet network.
- Slow-control capability is realized with a dedicated UART / RS232 interface.
- Both on-board DDR2 SRAM and a DDR2 SDRAM socket provide fast memory for data buffering and online processing.
- A large number of single-ended I/O signals accessible via mezzanine connectors can be used for interfacing slow-speed electronics.

Figure 7.27 (a) and (b) show a photograph and the block diagram of the MPRACE-2 board, respectively.

**10GbE / PLL Mezzanine** The development of the 10GbE / PLL mezzanine has been triggered by the need to supply the MPRACE-2 with additional capabilities and prepare it for emulating basic PPT functionality.

While the MPRACE-2 can be used as-is for receiving high-speed data from the IOB, it lacks a dedicated optical 10GbE interface for further data transmission towards a PC. On the other hand, the board supplies a sufficient number of high-speed MGT lanes that can be used to attach an external 10G PHY. The mezzanine realizes this approach with a 10G PHY of type VSC8486 from Vitesse, which connects to the MPRACE-2 FPGA using four full-duplex 3.125 Gb/s MGT lanes and the XAUI protocol. An SFP+ transceiver module interfaces with the PHY and provides the

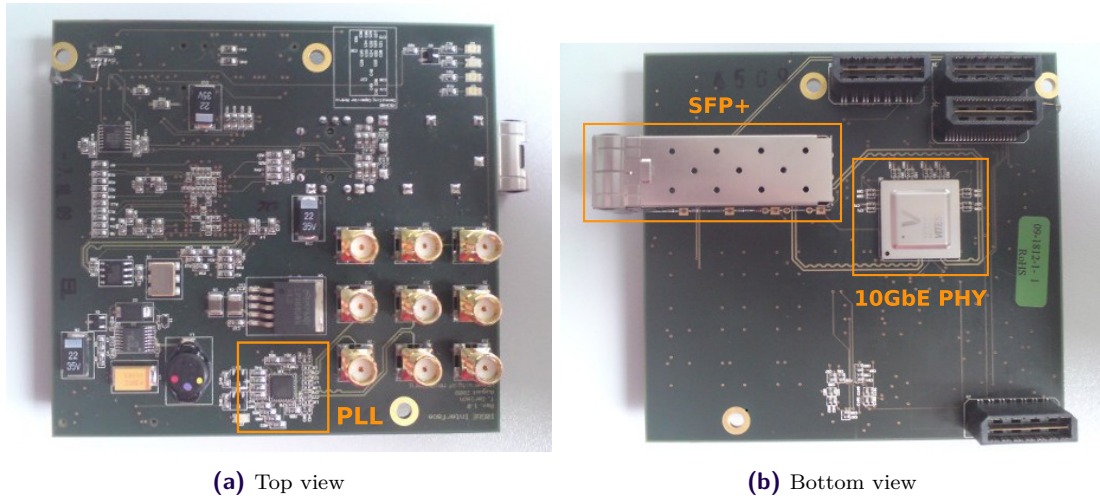




**Figure 7.27:** Photograph (a) and block diagram (b) of the MPRACE-2 FPGA board. Source: [40]

optical 10GbE link towards a PC equipped with an 10GbE network card for data taking. Additionally, the mezzanine houses a PLL of type ADF4350 from Analog Devices for evaluation purposes, as the initial draft of the PPT has foreseen to use this device for ADC clock generation. The PLL features internal registers which must be configured for proper operation of the device. The registers are accessible by the MPRACE-2 FPGA via a serial communication interface and can be programmed by either dedicated controller modules or software running on the MicroBlaze soft-core CPU.

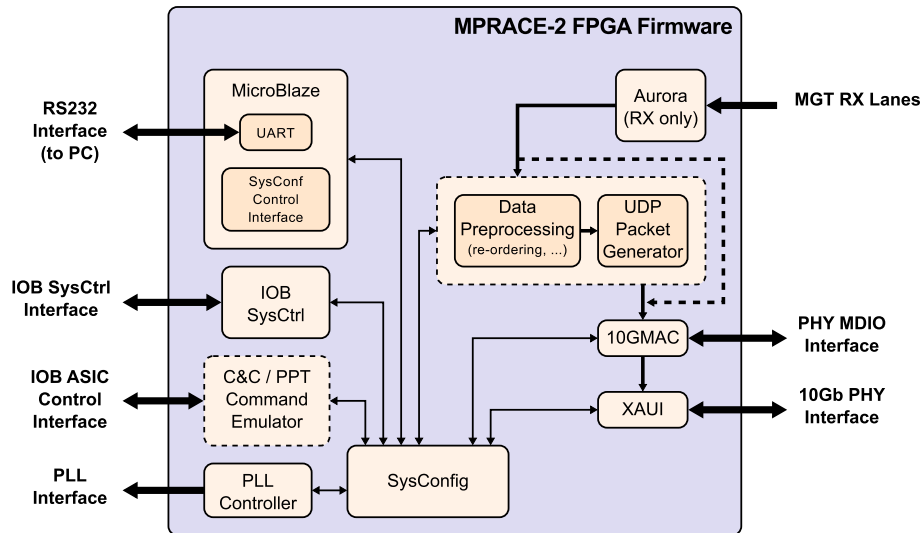
A photograph of both the top and bottom view of the 10GbE / PLL mezzanine is shown in figure 7.28 (a) and (b).



**Figure 7.28:** Top and bottom view of the 10GbE / PLL mezzanine for the MPRACE-2

### 7.3.2 MPRACE-2 FPGA Firmware

Figure 7.29 displays a block diagram of the MPRACE-2 firmware which has been developed to emulate PPT functionality as an initial testing platform for the IOB. The concept is similar to that



**Figure 7.29:** Block diagram of the MPRACE-2 firmware. Dashed components have not been fully implemented yet.

of the IOB firmware, as the functionality is distributed across several sub-modules. Central elements of the firmware are a SysConfig register bank and a MicroBlaze soft-core CPU. The register bank stores system-specific configuration data for the individual controller modules. An overview of the SysConfig memory map is given in chapter B.8. The microcontroller runs custom-developed software coded in C which provides a command line-based user interface for accessing SysConfig via a remote PC and an RS232 connection to operate the controller modules.

Slow-control of the IOB is realized using *IOB SysCtrl*, which act as master to the SysCtrl module of the IOB and implements the command protocol described in section 7.1.2 for communication.

The *C&C / PPT Command Emulator* is responsible for providing both proper bunch-synchronous timing and control commands to the IOB and ASICs, respectively. The present firmware version only features a draft implementation of this module.

The path for data transport from the MGT RX lanes to the optical SFP+ link comprises an *Aurora RX-only core* as well as a *10G MAC* and a *XAUI* core (all three developed by Xilinx), which provide the infrastructure for 10G Ethernet communication. Although the modules have been individually simulated and verified, the link between the Aurora and the 10G MAC is not fully implemented yet. Also, the present firmware implementation lacks the data pre-processing unit and the UDP packetizer.

Configuration of the PLL device on the mezzanine is managed by the *PLL Controller*, which implements the mechanism for programming the internal device registers. The configuration data are provided directly by the software via SysConfig.

### 7.3.3 MPRACE-2 FPGA Software

The standalone software running on the MicroBlaze CPU represents a user interface that conveniently allows testing the IOB electronics. Using a software-based approach for providing system configuration data has the advantage of flexibility, as the code can be adjusted more easily to different testing conditions than the hardware logic. The software is written in plain C, and its development is driven by the idea to offer a scalable code framework. That is, the software should be extendable in a straightforward manner whenever new functionality is required. Therefore, a plugin-based approach has been chosen. The functionality is implemented by different commands, which can be hooked into a console-like command interpreter. The commands access corresponding SysConfig address spaces of both the MPRACE-2 and the IOB for configuring and operating the system modules. Configuration data are provided via various command parameters and options. Figure 7.30 shows a screenshot of the software running on the MPRACE-2. The control commands supported by the latest software revision are listed in table 7.3.

```

Welcome to MPRACE-2 00.
help
Available commands:
  help  List the available commands, and describes their options.
  ver   Displays hardware/software version information.
  clr   Enables access to the IOB CLR controller
  iobsysconf Grants access to IOB SysConf register bank.
  iobver Displays hardware version information of IOB.
  lmk   Enables access to the LMK.
  pll   Enables access to the mezzanine PLL.
  prb   Enables access to the PRB.
  sysconf Grants access to SysConf register bank.
iobver
Hardware version:
SVN revision: 264
SVN status: Modified locally
Build date: 2013-02-21 11:22

```

**Figure 7.30:** Screenshot of the latest MPRACE-2 software.

Command	Option	Description
exit	-	When running on Linux, exit program Otherwise do nothing
help	[cmd]	Display help screen, or help to specific command
ver	-	Display local version information
fet	on   off	Enable/disable FET drivers
	delayon   delayoff	Get/set various delays
iobsysconf	read   write	Dedicated read/write access to IOB SysConfig
iobver	-	Display IOB version information
lmk	init	Initiate LMK programming
	devsel   data	Get/set device selection bitmask and configuration data
pll	on   off   mute   unmute	Power-on/off and mute/unmute PLL
	read   write	Read/write configuration data
	init	Initialize device registers R5 to R0
prb	on   off   reset	Enable/disable/reset PRBs
	bitmask   gdpsdelay	Get/set bitmasks and GDPS delay
sysconf	read   write	Dedicated read/write access to SysConfig

**Table 7.3:** Commands being supported by the latest MPRACE-2 software.

The present implementation of the software communicates with a remote control PC over UART / RS232 instead of the 1GbE interface provided by the MPRACE-2. Implementing a full-featured

TCP/IP communication with the main FPGA would have included to additionally design firmware for the MPRACE-2 bridge FPGA, which the on-board 1GbE PHY is connected to. Both Ethernet functionality and communication with the main FPGA over the serial high-speed transceiver links must then be provided by the bridge device firmware. On the other hand, solely using the bridge FPGA for testing the IOB is not an option, as the mezzanine connectors required are accessible only by the main FPGA. Considering the fact that MPRACE-2 is foreseen as a first testing platform only and will be replaced by the PPT prototype, the approach using the RS232 protocol has been chosen, as there is no need for high-speed Ethernet communication in IOB prototype testing.

The command line-based RS232 approach has the advantage of straightforward scripting capability, that is, a sequence of commands can be saved in human-readable text files and be transmitted via console from the remote control PC. In that way, different presettings can be applied depending on the test purpose or test environment. Scripting also allows for automatic runs of calibration and system checks. Additionally, a graphical application front-end could serve as user interface for a more convenient management of the command files. For example, various delay values could be adjusted by software slide bars, with the present values being stored as parameters in a command file. Likewise, the configuration settings for the Main Board clock buffers could be listed as a well-arranged table. Scripting capability thus provides manual and automatic control at the same time.

## 8 Signal Analysis and System Verification

For reliable operation of the DSSC DAQ system, it is indispensable to subject the developed hardware and the firmware logic to intense examination. The tests were performed with the setup described at the end of the previous chapter, which uses the MPRACE-2 as host board to remotely control the IOB. Additionally, some of the checks were exercised under semi real-life conditions with prototype hardware of the sensor Main Board, the PRB, and the MIB.

For the measurements, the following equipment has been used:

- LeCroy SDA 816zi-A, a four-channel digital storage oscilloscope with 16 GHz bandwidth, 40 GS/s sampling rate, and an integrated serial data analyzer
- Agilent 81133A, a 1.3 GHz pulse pattern generator with  $< 2$  ps rms clock jitter

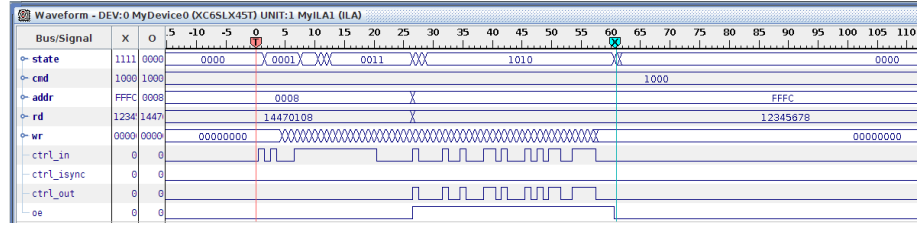
This chapter summarizes the results of the signal measurements and firmware tests for the individual electrical interfaces and firmware modules. A note on the ripples that occur on the oscilloscope signals throughout the measurements shown in the following. They originate from a sub-optimal ground connection of the measuring probe, as the default probe ground path is of about 10 cm in length. By using a shorter ground connection, the ripples can be reduced.

### 8.1 Slow-control Interface

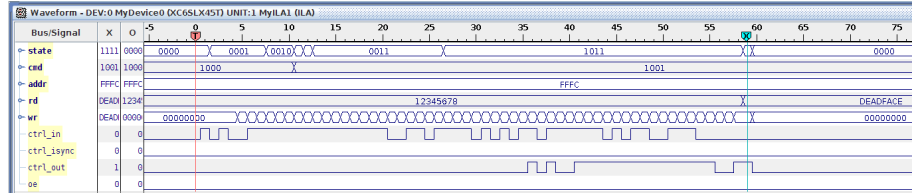
The functionality of the IOB slow-control interface has been verified at 20 MHz operation frequency. ChipScope captures of both read and write access to the IOB SysConfig from the MPRACE-2 are presented in figure 8.1 (a) and (b), respectively. The figures show the transactions from IOB side view and verify the proper reception of the serial protocol bits over signal `CTRL_IN`. Both start sequence and transaction command as well as the address of a SysConfig register (here `0xFFFC` of the dummy register) are transmitted first. Depending on the type of access, the 32-bit wide register content (`0x12345678`) is serially sent back over `CTRL_OUT` (read access), or another 32 bits (`0xDEADFACE`) are received over `CTRL_IN` (write access).

The results of signal analysis are shown in figure 8.2 (a) and (b). The yellow curve shows the transmission clock, which is generated by the MPRACE-2 and runs at a frequency of 20 MHz, while the purple curve illustrates the protocol bits transmitted over the data line.

The test results show proper functionality of the SysCtrl module and verify the signal integrity of the slow-control interface. The remote control mechanism allows the SysConfig register bank to be accessed by external master electronics (here MPRACE-2), which has been done throughout the entire testing phase proving a reliable communication between the test system components.



(a) Read access



(b) Write access

**Figure 8.1:** ChipScope capture of SysConfig (a) read and (b) write access by SysCtrl module via MPSPACE-2.

## 8.2 ASIC Control Interface

Controlling the ASIC readout is with the responsibility of the PPT, which provides command telegrams through the ASIC control interface routed across the IOB PCB. The interface signals ADCCLK, XCLK, and XDATA must ensure signal integrity at frequencies of 695 MHz, 99 MHz, and 4.5 MHz, respectively. In figure 8.3, the measurement of signal ADCCLK is shown, which has been fed with an 800 MHz clock signal<sup>1</sup> on the IOB TOLC connector at PPT side. The output clock signal on the IOB SOLC connector towards the sensor Main Board was probed and is represented by the green curve (oscilloscope channel C4).

The transmission quality of the traces for transmitting the ASIC command clock and ASIC command telegrams has been verified by applying a 156 MHz LVDS clock signal to both XCLK and XDATA on the TOLC input. The output signal measured at the SOLC connector is shown in figure 8.4, which looks similar for the two signal paths.

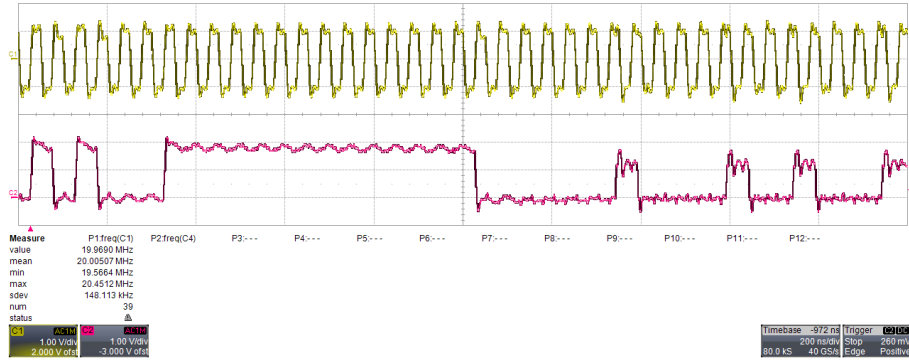
As the fourth ASIC control line **XRESET** (global detector FEE reset) does not need to fulfill demanding requirements on the transmission characteristics, its electrical connectivity was tested only and confirmed positively.

The measurements performed on the four fast control lines verify signal integrity of the ASIC control interface.

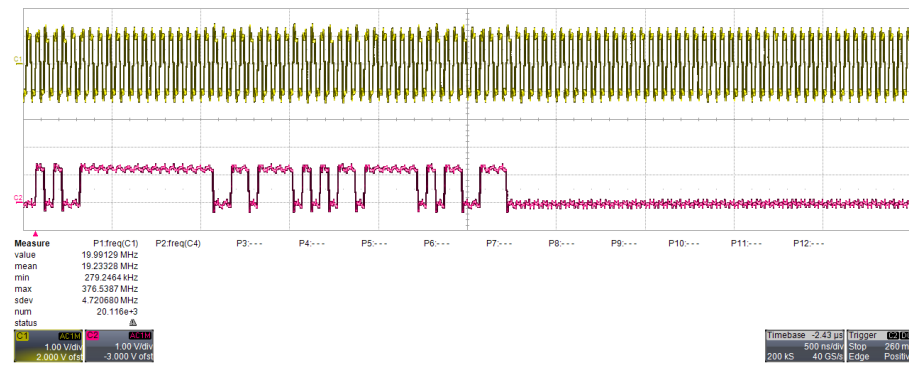
## 8.3 ASIC Data Readout

To verify signal integrity of the ASIC data input lanes ASIC\_D0<1..16> at the targeted data rate of 350 Mb/s, the IOB FPGA was configured to output a 350 MHz LVDS clock signal via those lines. The results of the measurement are presented in figure 8.5 (representative).

<sup>1</sup>Initially foreseen when the XFEL FEE clock was defined to 100 MHz

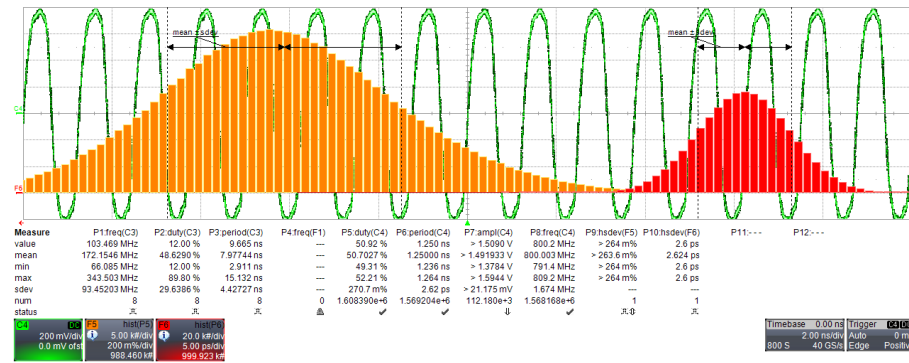


(a) Read access



(b) Write access

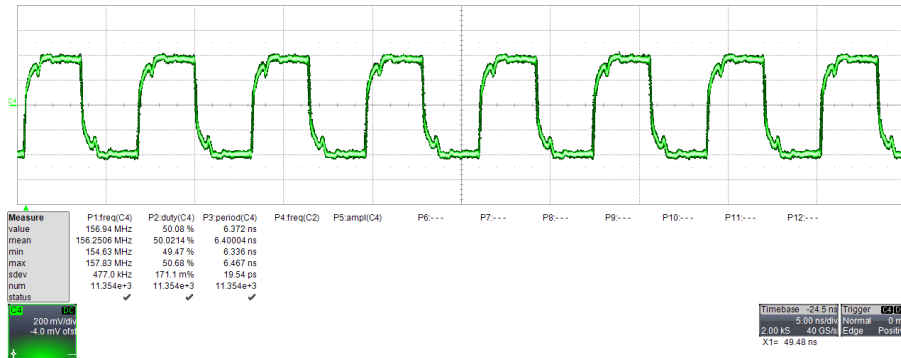
**Figure 8.2:** Screenshots of signal analysis of both clock (yellow) and data (purple) signals of IOB slow-control interface. Sub-figure (a) shows a zoom of read-access protocol transmission, while sub-figure (b) shows the full protocol transmission of a write transaction.



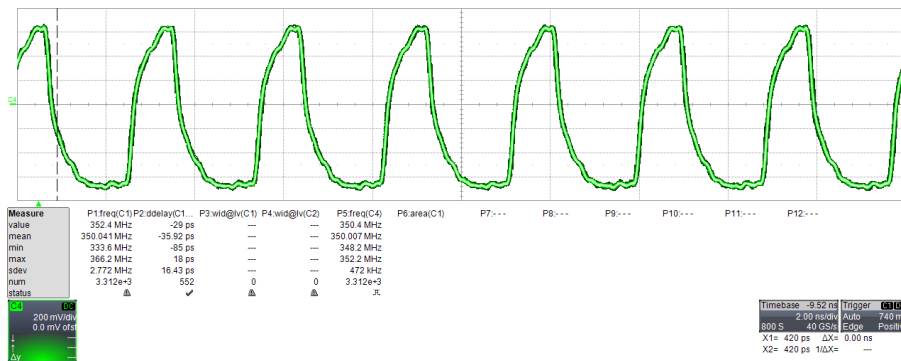
**Figure 8.3:** Signal measurement of the ADC clock at the IOB-to-Main Board interface (SOLC). The result shows an excellent 800 MHz clock signal.

The functionality of the AsicRoCtrl module was tested by providing a serial input data stream at different bit rates. The bit stream was generated by the MPRACE-2 and repeatedly transmitted at 156 Mb/s and 350 Mb/s. Figure 8.6 illustrates two waveforms captured with ChipScope analyzer. While the test pattern 0xDEADFACE is properly captured at a transmission rate of 156 Mb/s (a),

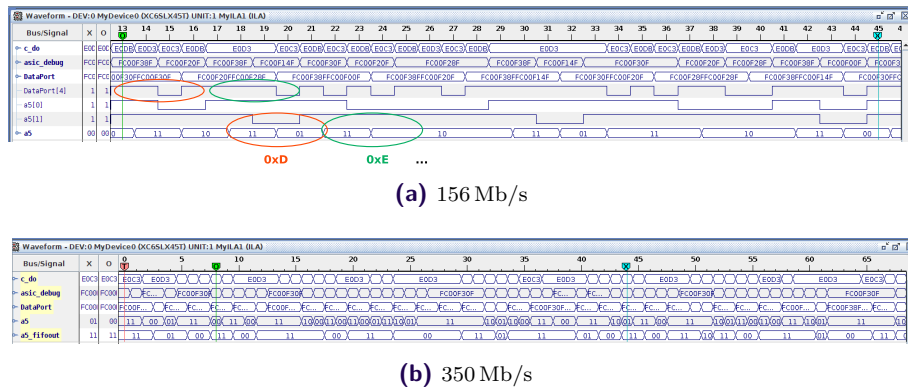




**Figure 8.4:** Measurement of signal quality of XCLK trace at 156 MHz. The signal waveform, which is similar for the XDATA trace, confirms the signal integrity of the two signals.



**Figure 8.5:** Signal measurement of data lane ASIC\_D05 (representative) of the IOB. The FPGA was configured to output a 350 MHz LVDS clock signal. The captured signal verifies the signal integrity.



**Figure 8.6:** ChipScope waveform of AsicRoCtrl test on data input channel ASIC\_D05. At a rate of (a) 156.25 Mb/s, the repeatedly transmitted 32-bit data (0xDEADFACE) are properly captured. When increasing the rate to (b) 350 Mb/s, bit errors occur.

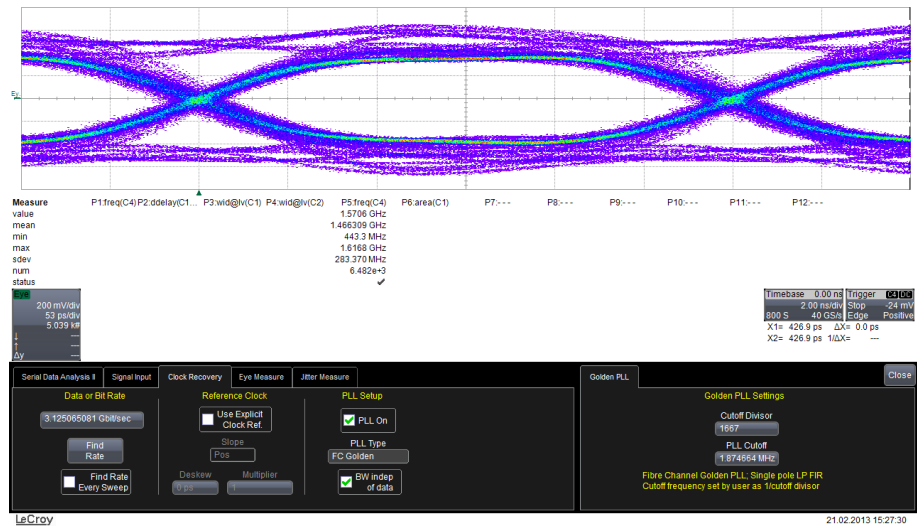
the SerDes mechanism seems to fail when increasing to 350 Mb/s (b). This is most likely related to sub-optimal board interfacing, as recent tests performed by the ASIC development group confirm successful data transmission at 350 Mb/s between their ASIC Test Board (ATB) used as data source



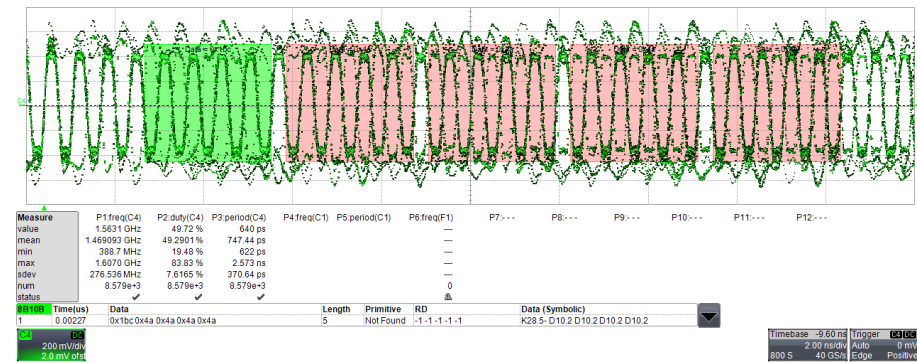
and the IOB.

8.4 High-speed Transceivers

Signal integrity of the IOB high-speed transceiver links has been verified by the measurements shown in figure 8.7. An Aurora TX-only core has been instantiated running at a data rate of 3.125 Gb/s.



(a) Eye diagram

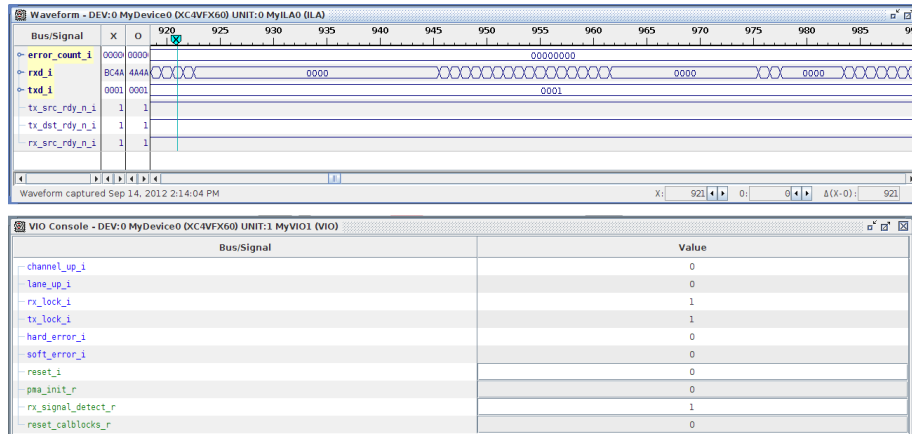


(b) K-character recognition

**Figure 8.7:** Signal measurement of GTP TX channels running with Aurora protocol. The eye diagram in figure (a) has been captured at a data rate of 3.125 Gb/s and confirms a very good signal quality. Figure (b) shows the proper transmission of 8B/10B encoded synchronization characters (K28.5-D10.2-D10.2-D10.2-D10.2 or 0xBC4A4A4A) as generated by the Aurora TX core of the IOB.

The eye diagram shown in sub-figure (a) confirms a very good signal quality. Additionally, picture (b) illustrates the proper recognition of 8B/10B encoded synchronization patterns generated by the Aurora TX core.

The next step was to establish a reliable high-speed data connection between the IOB and the MPRACE-2 using the GTPs and the Aurora protocol. However, the receiver was not able to complete the synchronization phase and properly lock on the sync characters provided by the transmitter, as shown in figure 8.8. Possible reasons for the issue have been narrowed down to:

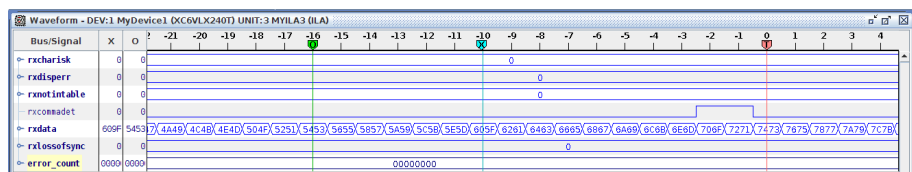


**Figure 8.8:** ChipScope waveform of MPRACE-2 Aurora RX core. The Aurora RX core is not able to properly finish the synchronization phase.

- Bit errors that occur on a specific receiver lane of the MPRACE-2.
- Violation of signal integrity due to poor electrical connections on the interfaces between the IOB and any of the receiver boards.
- Abortion of synchronization sequence by the TX core due to timeout violations.

The first assumption was disproved by observing the same issue when using Xilinx evaluation boards (ML605 and SP605) on the receiver side.

In order to rule out a violation of signal integrity, a Bit Error Rate (BER) test was performed using a standard Xilinx *GTP Transceiver Wizard* core. Unlike the Aurora core, this core does not implement a high-level transmission protocol and uses basic 8B/10B comma alignment for synchronization. Data were transmitted at a rate of 3.125 Gb/s per lane. As shown in figure 8.9, no errors occurred during 21 hours of operation, yielding a BER less than  $10^{-14}$ . The BER test



**Figure 8.9:** BER performed on the GTPs of the IOB FPGA. During 21 hours of operation at 3.125 Gb/s per data lane, no errors have occurred, yielding a BER  $< 10^{-14}$ .

completed successfully.

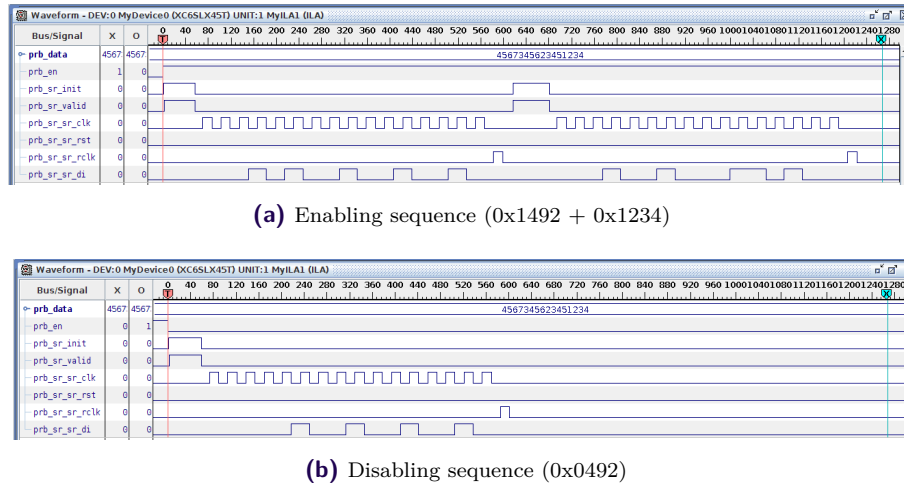
As the ChipScope capture in figure 8.8 indicates, a timeout issue is the most probable reason that the RX core fails to lock. After approximately 15  $\mu$ s, the TX core interrupts sending synchronization patterns and starts transmitting a few zero-data characters (0x0000 or D0.0 in 8B/10B coding). Then it restarts transmission of synchronization patterns until it interrupts again. However, recent

test results acquired by the ASIC development group confirm a successful transmission of high-speed data via the GTP lanes. The test setup is based on a combination of the IOB, the ATB, and a slightly different Aurora configuration. While the Aurora cores used in the tests presented above uses a dedicated handshaking interfaces – the so-called back channels – for initialization and maintenance, the cores of the ATB setup make use of a timer-driven controlling mechanism (cf. [56], p. 53f.).

The measurements and BER tests performed on the IOB GTPs confirm the signal integrity of the high-speed transmission lanes. As recent test results have shown, data transmission with the Aurora protocol is possible and will be further investigated.

## 8.5 PRB Control Interface

The ChipScope captures presented in figure 8.10 (a) and (b) verify the functionality of the PrbCtrl module. The figures show the serial transmission for a single PRB only. The transmission clock

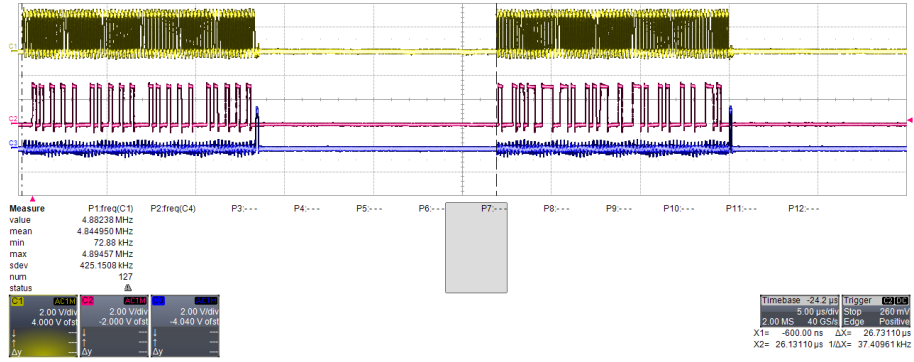
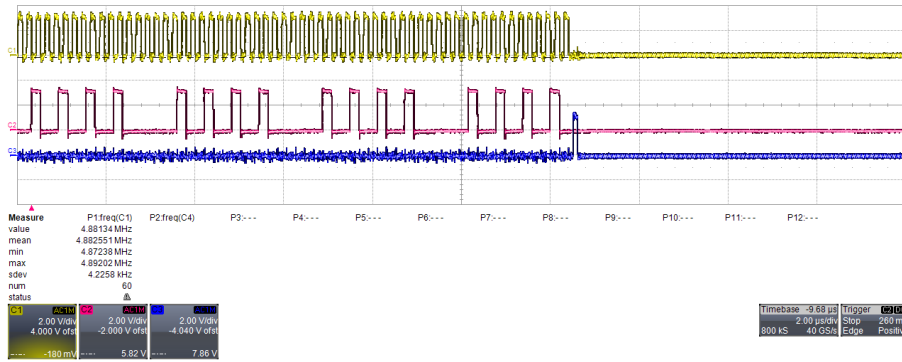


**Figure 8.10:** ChipScope capture of PRB shift register programming sequence for (a) enabling and (b) disabling the ASIC supply voltages. This view shows the transmission of 16 bit only, corresponding to the operation of a single PRB.

to the PRB shift registers is provided via signal PRB\_SR\_SR\_CLK, while serial data and the latch enable signal are sent over PRB\_SR\_SR\_DI and PRB\_SR\_SR\_RCLK, respectively.

The enabling phase is triggered on rising edge of the controller enable signal PRB\_EN and first transmits the GDPS bitmask (0x1492), followed by an arbitrary power net activation bitmask of 0x1234. The falling edge of PRB\_EN signal causes the controller to transmit the bitmask 0x0492, which disables all power nets but power net 2. The bitmasks are stored in the shift registers by asserting signal PRB\_SR\_SR\_RCLK.

Figure 8.11 (a) and (b) show the signal measurement of the ASIC power enabling and disabling sequence, respectively, for four daisy-chained PRBs. That is, each transmission cycle comprises four subsequent 16-bit words. The transmission clock signal (about 4.9 MHz) is represented by the yellow curve, while the data and the latch enable signal are represented by the purple and blue curve, respectively.

(a) Enabling sequence ( $4 \times 0x1492 + 0x4567.3456.2345.1234$ )(b) Disabling sequence ( $4 \times 0x0492$ )

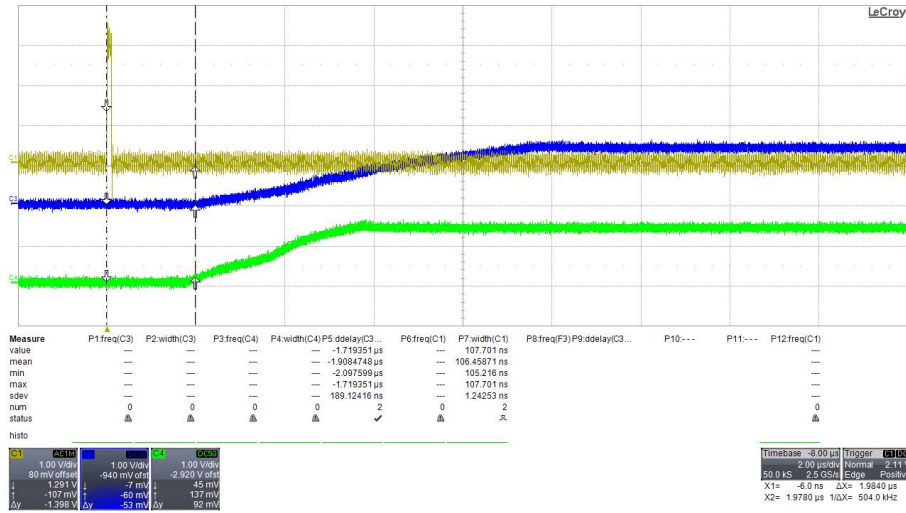
**Figure 8.11:** Measurement of the control signals generated by PrbCtrl for operating the ASIC supply voltages.

When activating the ASIC power nets, the GDPS enable pattern  $0x1492$  is sent four times in a row. The transmission is completed by asserting the latch enable signal. The GDPS delay has been set to 4096 156.25 MHz cycles (about  $26.2 \mu s$ ), which is also verified by the graphically estimated  $\Delta X$  value of  $26.7 \mu s$  in sub-figure (a). Next, the four arbitrary bitmasks  $0x4567$ ,  $0x3456$ ,  $0x2345$ , and  $0x1234$  for PRB are transmitted and stored by asserting the latch enable signal. The daisy-chaining of the regulator boards requires sending the bitmasks in descending PRB order.

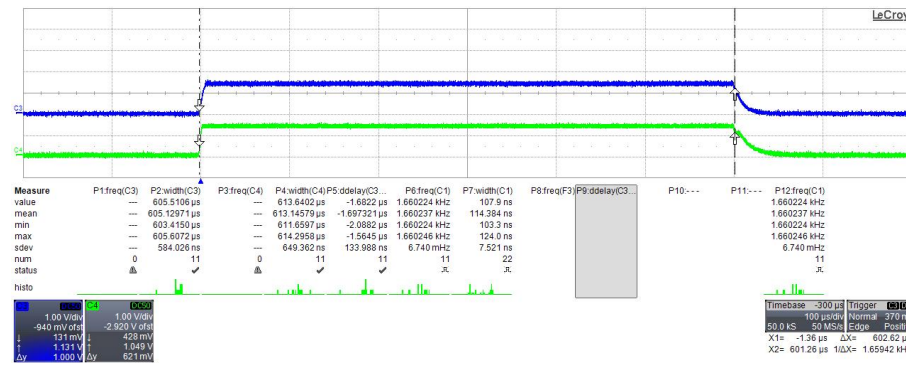
For de-activating the ASIC supply voltages, the disabling bit sequence  $0x0492$  is sent four times in a row. This special pattern disables all power nets but the digital power of net 2, as it is required for data readout during the inter-train gap.

The results of operating a single PRB prototype using the IOB test environment are presented in figure 8.12. Sub-figure (a) shows the delay between the shift register latch enable signal (yellow) and the time the voltage start to rise. It is roughly about  $2 \mu s$  (cf.  $\Delta X$ ). After another  $10 \mu s$ , both power net 1 (green) and power net 3 (blue) have stabilized. Considering the preliminary ASIC power timing presented in chapter 7.1.3, the enabling sequence must be started about  $14 \mu s$  prior to the first incoming bunch. In picture (b), a complete  $602 \mu s$  power cycle of power net 1 (green) and 3 (blue) is shown.

The results of the measurements presented above verify both functionality of the PrbCtrl module and signal integrity of the IOB electronics that interfaces with the PRBs.



(a) Power enable delay

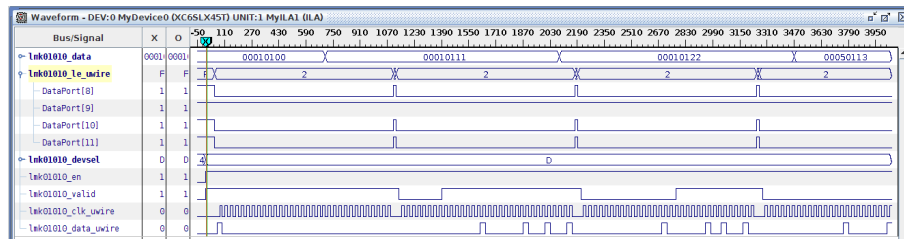
(b) 602  $\mu$ s power cycle

**Figure 8.12:** Measurement of ASIC power cycling test. Sub-figure (a) shows the delay of shift register latch enable signal (yellow) to start of rising power nets 1 (green) and 3 (blue). The measurement of a complete 602  $\mu$ s power cycle is illustrated in sub-figure (b).

## 8.6 Main Board Clock Buffer Interface

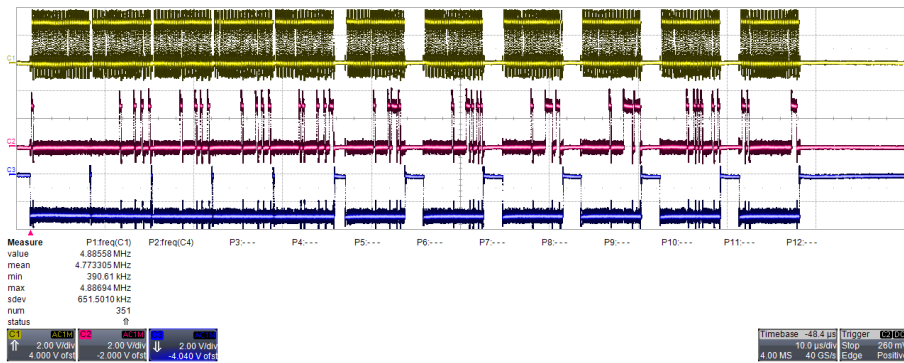
Proper operation of the LmkCtrl is confirmed by the ChipScope recording shown in figure 8.13. When asserting the controller enable signal LMK01010\_EN, the 32-bit data assigned to data bus LMK01010\_DATA are transmitted serially on signal line LMK01010\_DATA\_UWIRE. The transmission clock is provided by LMK01010\_CLK\_UWIRE. According to the recommendations given in the clock buffer data sheet [55], the device is reset by first writing to configuration register R0. Subsequently, the actual configuration data are written to the device registers. During the serial transmission, the low-active latch enable signals LMK01010\_LE\_UWIRE<n> are asserted according to the 4-bit device selection mask (here 0xD). Each transaction of a 32-bit data word is finished by de-asserting the LE signals, and the next data word is fetched and transmitted as long as the controller is enabled and data are marked valid.

Integrity of the clock buffer interface signals is verified in the measurement results shown in figure 8.14.

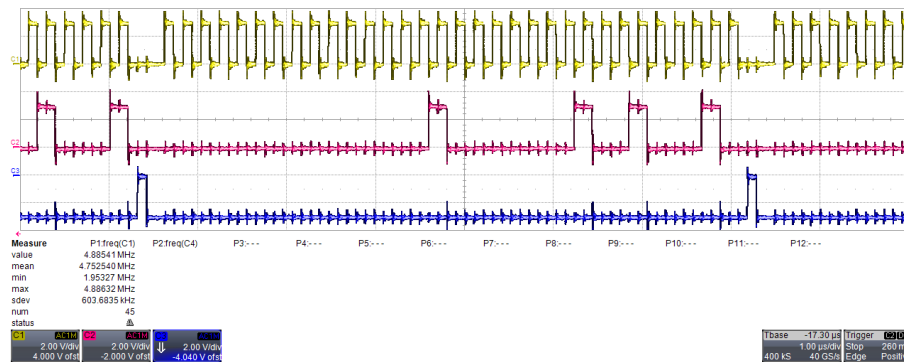


**Figure 8.13:** ChipScope recording of LMK01010 programming sequence provided by the LmkCtrl module. Prior to the first actual configuration data word assigned to the 32-bit input data bus, the clock buffer device is reset by writing register R0, which is recommended in the device data sheet [55]. During the individual register programming phases, the low-active latch enable signals are asserted. The write transaction is finished by de-asserting the latch enable signals.

The yellow and purple curves display the transmission clock and the data bit stream, respectively.



(a) Complete cycle



(b) Zoom-in

**Figure 8.14:** Signal waveforms of (a) a complete LMK01010 initialization cycle and (b) a zoom-in. Channel 1 (yellow) is the transmission clock signal running at a frequency of about 4.9 MHz. The transmitted data bits and the clock buffer latch enable signal are shown by the purple and blue curve, respectively.

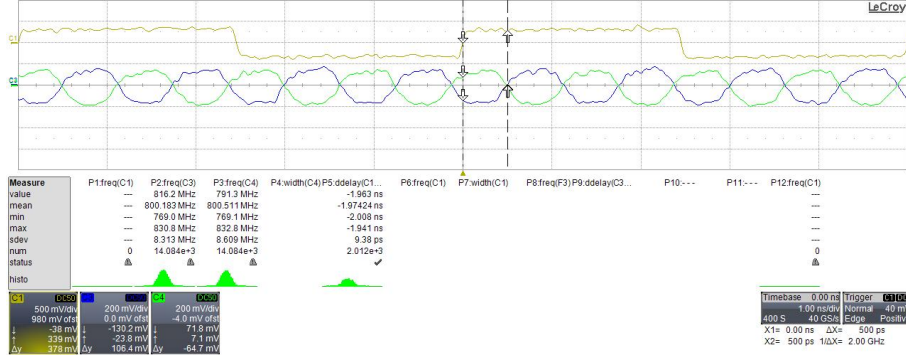
A representative of a latch enable signal is shown by the blue curve.

In sub-figure (a), a complete configuration cycle was captured, which comprises writing to all 10 device registers. A zoom-in of the configuration sequence is given by sub-figure (b), which shows

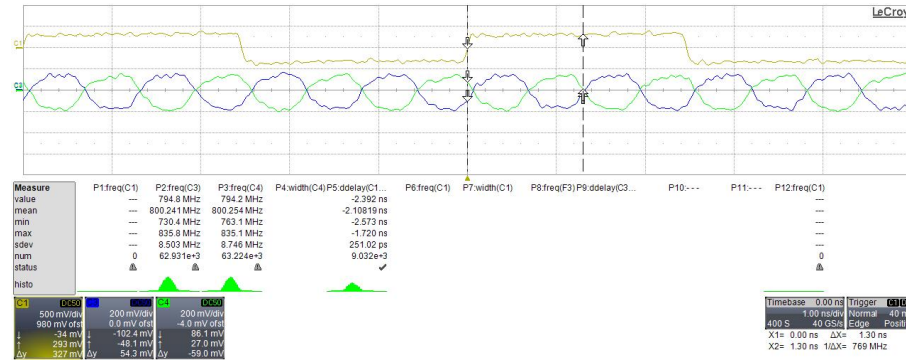


that serial data are asserted on the falling edge of the transmission clock, as implemented in the controller FSM.

The LmkCtrl has additionally been tested on a sensor Main Board prototype, which provides a single LMK01010 clock buffer device. The test was primarily related to elaborate the delay mechanism of the clock buffers. The results are presented in figure 8.15. The clock buffer was



(a) Bypassed delay stage



(b) 450 ps delay

**Figure 8.15:** Input-to-output delay measurements of Main Board clock buffer controlled by LmkCtrl.

provided with an 800 MHz input clock signal. Both the positive (green) and negative (blue) output clock signal as well as the reference input signal (yellow) have been recorded, and the delay was measured. The yellow curve actually shows a 200 MHz signal (reference clock signal divided by four), which is due to a limited test setup. The second differential clock output of the frequency generator at hand did support 300 MHz clock signals only. However, this was no issue for the delay test.

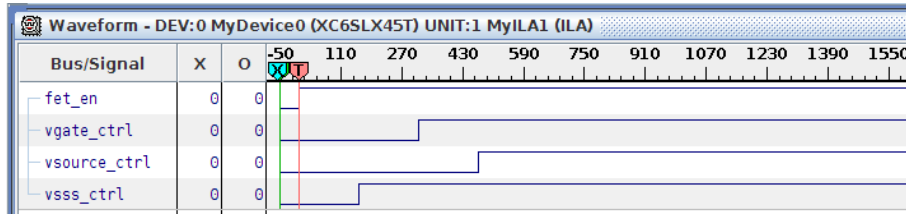
The first picture (a) shows the output clock signal of the LMK01010 device configured to bypass the internal delay stage. The graphically estimated input-to-output delay of  $\Delta X = 500$  ps is the device-specific internal gate delay. In the second picture (b), the clock buffer was programmed to add a delay of  $\Delta t = 450$  ps. However, the internal delay stage additionally infers an offset of about 400 ps. In total, the input-to-output delay is calculated to 1350 ps, which is verified by the graphically estimated value  $\Delta X = 1.3$  ns.

The different test performed on the IOB prototype and the LmkCtrl module conclude a proper

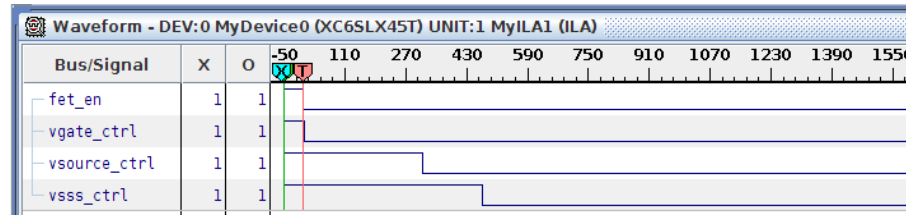
operation of the programming mechanism. Also, the clock buffer delay test verifies that the LMK01010 device suits the demand for compensating for signal skews in the sub-nanosecond range.

## 8.7 FET Driver Control Interface

The ChipScope waveforms of figure 8.16 show the FET driver control signals generated by FetCtrl. The controller was driven by the 156 MHz fabric main clock. Controller configuration (i.e. delay



(a) Enabling sequence



(b) Disabling sequence

**Figure 8.16:** ChipScope capture of (a) enabling and (b) disabling sequence generated by the FetCtrl module. The controller was operated via MPSPACE-2 software.

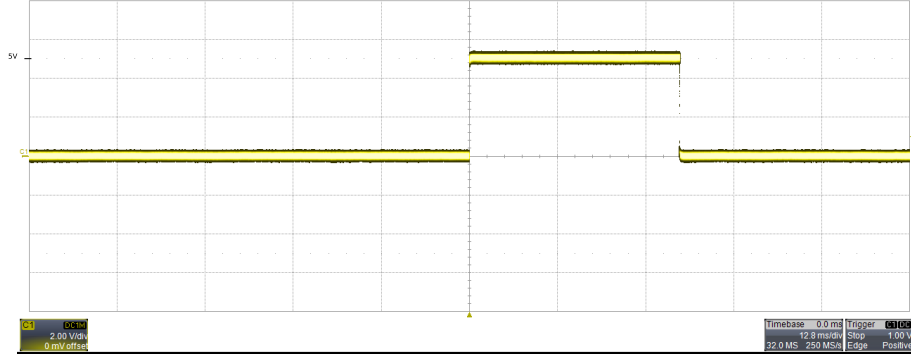
values of the individual voltage control signals) and operation have been managed using the MPSPACE-2 software. The delay values have been set as follows:

- $d_{vgateon} = 2 \mu s$  (approx. 312 clock cycles)
- $d_{vsourceon} = 3 \mu s$  (approx. 468 clock cycles)
- $d_{vssson} = 1 \mu s$  (approx. 156 clock cycles)
- $d_{vgateoff} = 0 \mu s$  (0 clock cycles)
- $d_{vsourceoff} = 2 \mu s$  (approx. 312 clock cycles)
- $d_{vsssoff} = 3 \mu s$  (approx. 468 clock cycles)

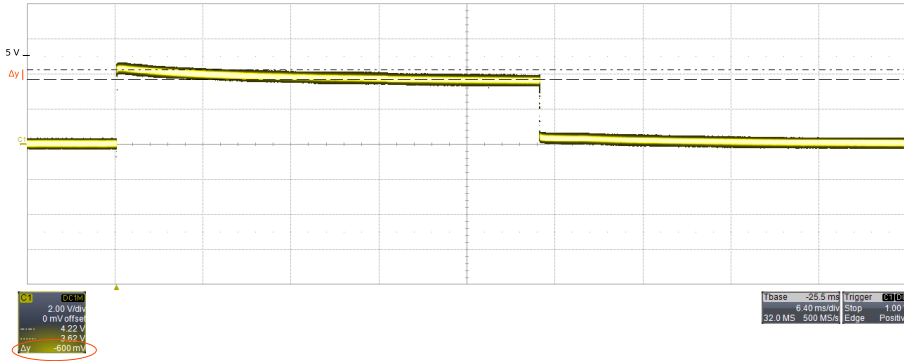
On rising edge of FET\_EN, the delay counters are loaded with the corresponding enable delay values, which is reflected in the order the voltage control signals VGATE\_CTRL, VSOURCE\_CTRL, and VSSS\_CTRL are asserted. A falling edge on the enable signal loads the disable delay values into the counters. Accordingly, the sequence for de-asserting the voltage control signals is different. The waveform also verifies the functionality of the delay mechanism, as indicated by the number of clock cycles between the rising edges of the enable signal and the control signals.



As the recent sensor concept foresees using  $V_{SSS}$  as ground reference, the electrical functionality of the circuitries for  $V_{SOURCE}$  and  $V_{GATE}$  has been analyzed only. The two circuitries were supplied with an input voltage of  $V_{in} = 5\text{ V}$ , which is well within the desired voltage ranges (cf. chapter 6.3.2). The results for the source voltage are presented in figure 8.17. Picture (a) shows the signal of the



(a) No output load



(b) Output load  $R_{load} = 1\ \Omega$

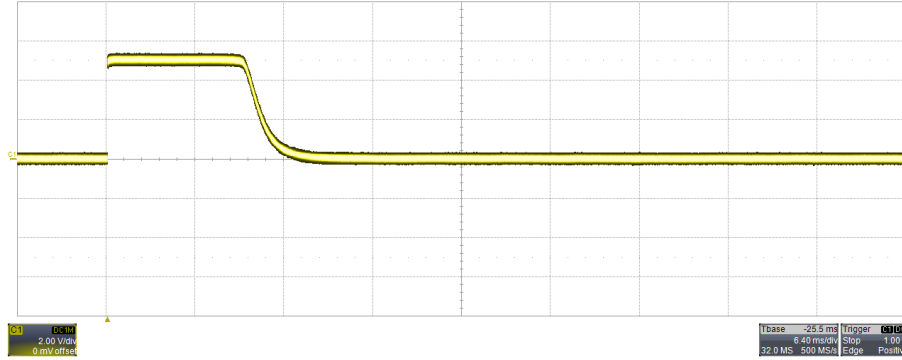
**Figure 8.17:** Gating of power net  $V_{SOURCE}$ . A reference input voltage of  $V_{in} = 5\text{ V}$  was supplied. Picture (a) shows the unconnected output voltage, which is  $V_{out} = 5\text{ V}$ . In picture (b), a dummy load of  $R = 1\ \Omega$  was added to the output branch. The maximum pulse width of the output voltage is limited by the RC circuit of the signal path of the FET driver control signal.

output voltage  $V_{out} = 5\text{ V}$  with no load provided, which is at the same level as the input voltage. In picture (b), a dummy load of  $1\ \Omega$  was inserted into the output branch of  $V_{SOURCE}$ . An output current of  $I_{out} \approx 4\text{ A}$  was observed. A rough graphical estimation shows that the output voltage has dropped by  $\Delta V_{out} \approx 1\text{ V}$  and further decreases by another  $\Delta v$  ( $\Delta y$ ) = 600 mV. The voltage drop  $\Delta V_{out}$  is mainly caused by the total source resistance of both transistor and PCB trace  $R_{tot} = R_{DS(on)} + R_{trace}$ . With a  $V_{SOURCE}$  trace geometry of  $W = 0.5\text{ mm}$ ,  $H = 18\ \mu\text{m}$ ,  $L \approx 12\text{ cm}$ , the trace resistance<sup>2</sup> is in the order of  $R_{trace} \approx 224\text{ m}\Omega$ . The transistor resistance is assumed to be about  $R_{DS(on)} = 10\text{ m}\Omega$  (estimation from data sheet). That is, a voltage drop of  $\Delta U = 936\text{ mV}$  is expected at the given output current, which agrees well with the measured value. The further signal development of  $V_{SOURCE}$  is related to the effect of contact resistance between the power supply and the voltage input nodes. The charged capacitors provide full power at the moment of gating, while the contact resistance limits the current flowing during the gating period,

<sup>2</sup>Specific electrical resistance of copper:  $\rho_{Cu} = 0.0168 \frac{\Omega \cdot \text{mm}^2}{\text{m}}$

and the output voltage decreases by  $\Delta v$ . The two pictures also show the effect of the RC circuit in the signal path of the FET driver control signal. The RC time constant limits the maximum pulse width to about 30 ms.

The result of the measurement performed on the open drain circuit of  $V_{GATE}$  is shown in figure 8.18. The picture shows the *inverted* signal development of the gate voltage applied to the p-channel



**Figure 8.18:** Inverted signal development of the gate voltage applied to the controlling p-channel MOSFET in the  $V_{GATE}$  circuitry.

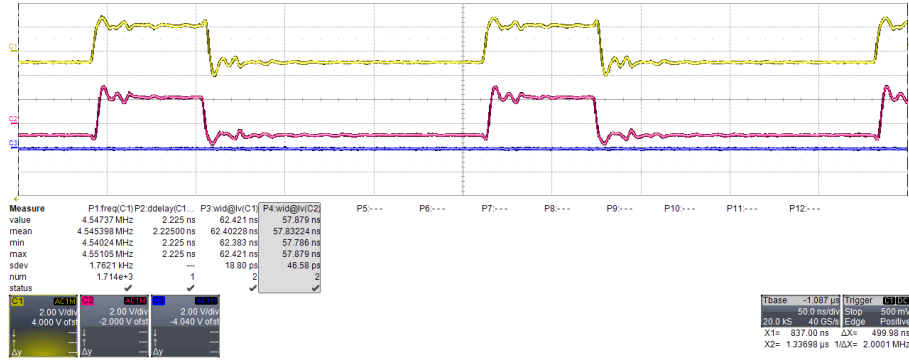
MOSFET, which operates the power net  $V_{GATE}$ . An unasserted control signal causes the gate voltage to remain at +5 V and closes the FET. When asserting, the gate voltage drops to GND, which opens the p-MOSFET and gates  $V_{GATE}$  to the output power net. The RC circuit located in the  $V_{GATE}$  control signal path limits the maximum pulse width to about 8.3 ms. After that period, the gate voltage rises back to +5 V, which shuts the p-MOSFET and cuts the power on the output side.

The results of the FetCtrl tests verify the functionality of both the firmware module and the switching electronics. The realization of the switching mechanism provides the flexibility (adjustable switching delays and safety measures) to meet the final requirements of the sensors.

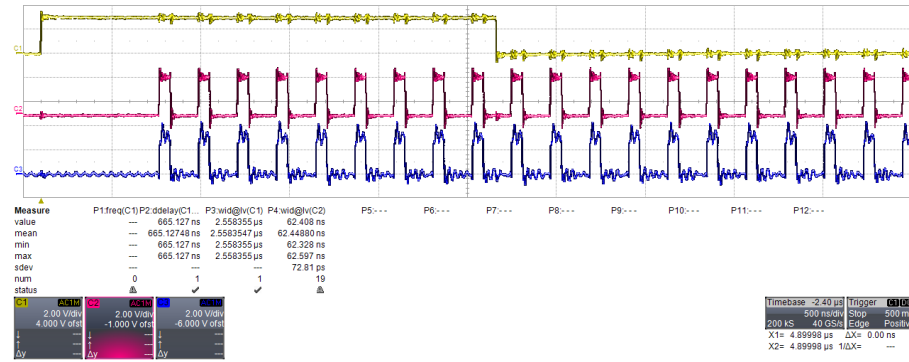
## 8.8 Sensor Clear Signal Interface

The mechanism for generating the clear signals is based on an implementation which uses the OSERDES primitives of the Spartan-6 FPGA. Due to the fact that the OSERDES data outputs are not able of being connected to the ChipScope interface (synthesis tool error), the available ChipScope recording is not presented, as it shows internal signal behavior only. Instead, both functionality of the ClrCtrl module and the signal integrity of the interface electronics have been verified by the measurements shown in figure 8.19 (a) and (b). The controller was clocked at a frequency of 100 MHz.

The first picture (a) illustrates the proper operation of the OSERDES-based delay mechanism, which allows for the characteristic relation between the CLR signal (yellow) and the CLRGATE signal (purple). The repetition rate of the CLR signal is 4.54 MHz, corresponding to 1/22 of the controller clock frequency. The OSERDESs have been configured to provide an edge-to-edge offset of one clock cycle in order to verify the minimum offset step size, which is 1/4 of the controller clock period (2.5 ns). The active period of CLR was set to six controller clock cycles, or 60 ns. Accordingly, the flat-top duration of CLRGATE should be in the order of 55 ns. The actual measurements show



(a) CLR-CLRGATE characteristic



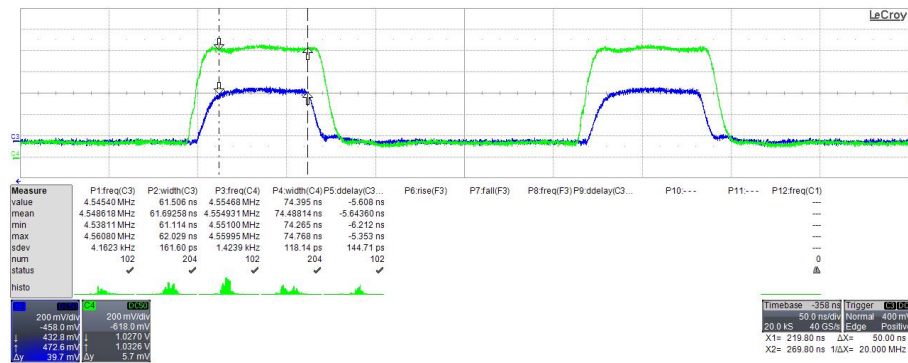
(b) Clear signal sequencing

**Figure 8.19:** Measurement of control signals generated by ClrCtrl module for operating the sensor clear signals.

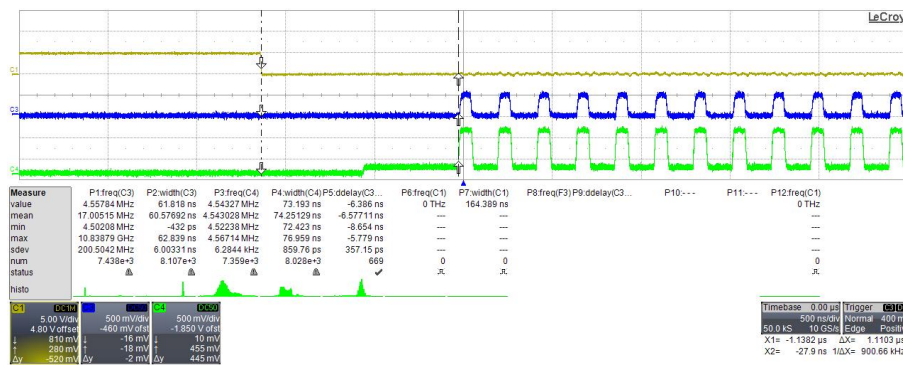
an edge-to-edge delay of about 2.25 ns, yielding a deviation of about 11 %. It is most likely related to the slew rate of the FPGA output signals. The deviation in the measured values of the flat-top periods (62.42 ns for CLR and 57.88 ns for CLRGATE) mainly result from the coarse x-locations of the measuring points at 90 % signal level.

The measurement shown in the second picture (b) is an example of clear signal sequencing. The yellow curve represents the  $\overline{\text{CLRDIS}}$  signal, while the purple and the blue curves show the CLR and CLRGATE signal, respectively. The measurement shows an offset between the rising edges of  $\overline{\text{CLRDIS}}$  and CLR which is in the order of 665 ns. The controller was configured for a delay of 64 controller clock cycles, or 640 ns. The discrepancy results from two FSM state transition, which have not been considered in the delay counter start value. For the flat-top period of  $\overline{\text{CLRDIS}}$ , a value of 256 controller clock cycles (2.56 ns) was programmed, which is perfectly confirmed by the measured value of 2.558 ns. The waveform also shows that the clear/clear gate pulsing remains active as long as ClrCtrl is enabled. As the clear disable signal actually controls the operation status of the clear signal gate drivers on the PRBs, the continuing clear/clear gate pulses have no effect when the gate drivers are disabled. However, it is the responsibility of the IOB master FSM to properly enable and disable the controller module.

In addition, the ClrCtrl has been tested in combination with a single PRB prototype. The clear signals generated by the gate drivers are shown in figure 8.20. Picture (a) illustrates the clear/clear gate relation, while picture (b) shows the beginning of the clear signal sequencing. The controller



(a) CLR-CLRGATE characteristic



(b) Clear signal sequencing

**Figure 8.20:** Sequencing of sensor clear signals generated by the clear signal gate drivers, which are operated by the ClrCtrl module. Figure (a) shows a close-up of the CLR-to-CLRGATE relation, while (b) shows a test sequence of clear signalling.

configuration was different from that of the test presented in the previous paragraph. Here, the edge-to-edge offset of the clear/clear gate signal is set to 5 ns. Also, the flat-top period of the clear signal is in the order of 70 ns. The measured values, however, show an additional slowing effect on the signals, which is induced by the gate drivers.

In summary, the tests and measurements performed on the ClrCtrl verify the concept and implementation of the clear signal generation. The electrical interface towards the PRB gate drivers is operational, and the integrity of its signals is confirmed.

## 8.9 Summary of Test Results

Various tests and measurements have been carried out to confirm proper operation of the IOB firmware modules and to verify signal integrity of the electronic interfaces.

The results of the signal measurements prove transmission characteristics that comply with the demands on the signal traces. Possible discrepancies observed on the traces of both the ASIC data input channels and the high-speed transceivers have been dropped by the results of recent experiments carried out by the ASIC development group.

The tests performed on the firmware confirm proper functionality of most of the custom-developed controller modules. Additionally, the results comply with those of the functional simulations presented in chapter B. Modifying the implementation of the AsicRoCtrl has enabled the ASIC development group to successfully capture and buffer simulated ASIC data on one channel at 350 Mb/s. By using a different configuration of the Aurora core, the ASIC group also verified high-speed data transmission using the Aurora protocol.

In summary, both the IOB prototype and the firmware modules, which have been developed within the scope of this thesis, represent an operational system that can be used for DSSC detector testing.



## 9 Conclusion and Outlook

In this thesis, the development of the DAQ front-end for the DSSC detector is motivated and presented. The work summarized in this document covers initial system concept studies and the development of a suitable hardware architecture that complies with the demanding requirements of the detector with regard to data bandwidth and timing. The resulting hardware architecture of the DSSC DAQ is a two-staged hierarchical structure that uses multiple instances of two FPGA-based modules – the Patch Panel Transceiver and the I/O Board.

Major subject of this thesis was the development of the I/O Board, including the operating FPGA firmware and a test environment for electrical and functional verification of the system. With the revised version of the I/O Board also presented in this document, the prototyping phase of the lower DSSC DAQ level is practically completed.

**Discussion and Conclusion** The development of both the I/O Board PCB layout and the FPGA firmware implied a lot of fundamental decisions with regard to the technology being applied and the general concept of data transfer and system control.

Considering the marginal feasibility of both a GPU-based implementation and readout modules that apply custom-built ASICs, using FPGA-based modules has the significant advantage of flexibility, while providing massive parallel data processing and at the same time keeping the number of employed devices at a minimum.

Due to the spatial limitations that come with the mechanical detector layout, full DAQ capability could not be implemented on a per-module level. On the other hand, critical detector electronics is required to be operated independently for each sensor module, which makes the effort to realize the DAQ functionality on a single detector-global module unreasonable. Hence, a hierarchical approach was chosen for the front-end DAQ system. The two-staged implementation allows separating global detector control (e.g. XFEL timing distribution and remote control) from module-specific peripheral controlling tasks.

At the time of starting the I/O Board PCB development, the Xilinx Spartan-6 family was the most reasonable choice to use with regard to device features and expense. Nowadays and about two years later, the Artix-7 family from Xilinx could provide another step in density optimization. The XC7A20SLT device has a size of only  $10\text{ mm} \times 10\text{ mm}$  (CPG237 package). Additionally, it features only a single high-speed transceiver at 6.6 Gb/s bandwidth, which would sufficiently cover the sensor module data volume and reduce the PCB routing effort.

The I/O Board is realized as a custom 10-layer printed circuit assembly with a 4-layer flexible Polyimide core, which was initially introduced to provide the movability of a detector quadrant. The flex core, however, induces a certain risk of damaging the copper traces when bending, especially on the edges of the rigid board sections. On the other hand, the latest detector layout combines the I/O Board along with the other system boards into a rigid quadrant arrangement, which minimizes the risk of broken traces. The ability to move the quadrant is given by flexible connection leads between the Module Interconnection Board and the quadrant plate and patch panel, respectively. Both prototyped and revised version of the I/O Board lack protection against overvoltage and

reverse power supply. This is one flaw which was not that obvious in the beginning and became lost of sight during the effort to achieve a most density-optimized DAQ component.

As for the FPGA firmware, the generic code implementation of some modules considerably eases adapting the system to different test environments. The concept of distributing the various controlling tasks to individual entities and providing local timing information through a dedicated master FSM allows the firmware being extended easily with additional features. For application in the final DSSC detector, however, more safety measures and error checking could be implemented to ensure a very reliable and failsafe system.

In summary, the DSSC front-end DAQ system uses latest developments in microelectronics wherever possible. In particular, the results of both signal analysis and firmware tests of the I/O Board confirm its capabilities with regard to the requirements.

**Outlook** The second revision of the I/O Board will presumably be available in spring 2013. Elaborate tests will conclude if the revised board can be applied in the final detector arrangement. On confirmation, a sufficient number of boards (i.e. at least 16 plus spare) is ordered, which will be assembled with the other detector components to gradually build the detector quadrants.

Major parts of the present I/O Board firmware can already be applied on initial detector test setups. However, its further development is an ongoing process that not only evolves with new revisions of detector hardware components during the development phase. The development of the firmware is also driven by the demand of user experiments to provide flexibility and new features for future applications.

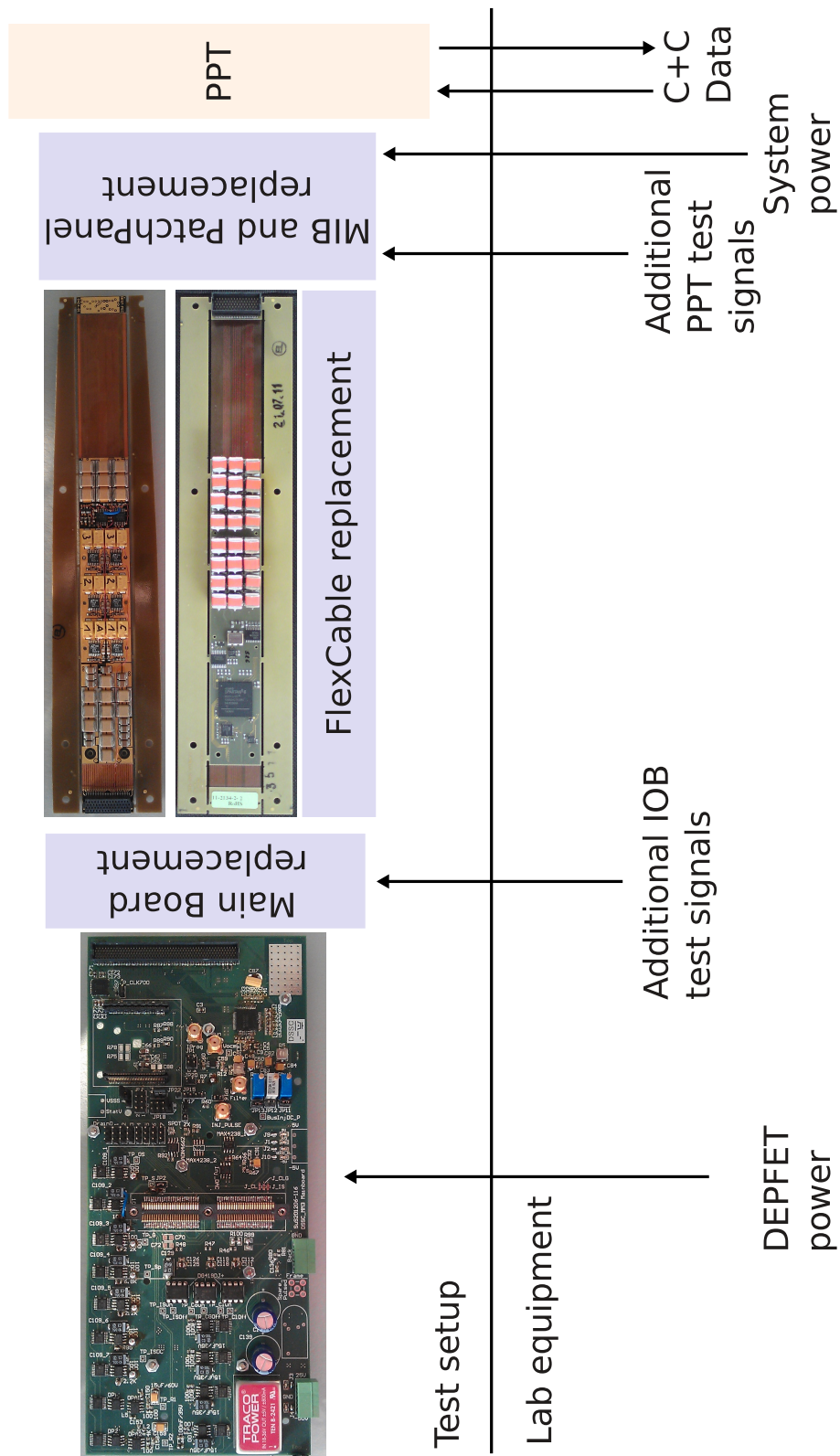
A major change that will be applied to the present I/O Board firmware addresses the concept of ASIC readout. The new readout approach will use a de-serialization mechanism with a SerDes ratio of 1:3. That is, the incoming data bits of a single ASIC are bundled in groups of three. This is possible, as the actual pixel data comprise 9 bits only, while the additional leading bit serves as start bit.

Another significant change will be the use of only three high-speed transceiver lanes for data transmission towards the Patch Panel Transceiver for technical reasons. Considering the estimated payload data rate of about 4.2 Gb/s per module, there will be no drawbacks in reducing the number of high-speed channels by one.

A prototype version of the Patch Panel Transceiver is expected to be available for testing in early spring 2013. By the time of writing this document, the development of Patch Panel Transceiver firmware has started. A counterpart to the IOB Aurora core has been implemented on the ASIC Test Board that enables for ASIC data readout tests at full transmission speed and will be ported to the Patch Panel Transceiver firmware. With the availability of the Patch Panel Transceiver prototype and its firmware, a DAQ test setup is planned as shown in figure 9.1. The system primarily serves for verification of the new I/O Board revision and the PPT hardware and for developing the software and firmware required for final detector operation. The test system will be controlled by a graphical software interface, which is currently being developed and will also serve as basis for the final operator environment of the DSSC system.

Testing a complete sensor module, that is a Main Board, four Power Regulator Boards, an I/O Board, and a Module Interconnection Board, is planned at the beginning of 2014. The test setup will also make use of the Patch Panel Transceiver and the graphical software environment.





**Figure 9.1:** Proposal for the DAQ test setup of a single module. The PCB shown on the left side is a replacement for the sensor Main Board. The brown and the green PCB shown in the middle are the Power Regulator Board and the I/O Board, respectively.



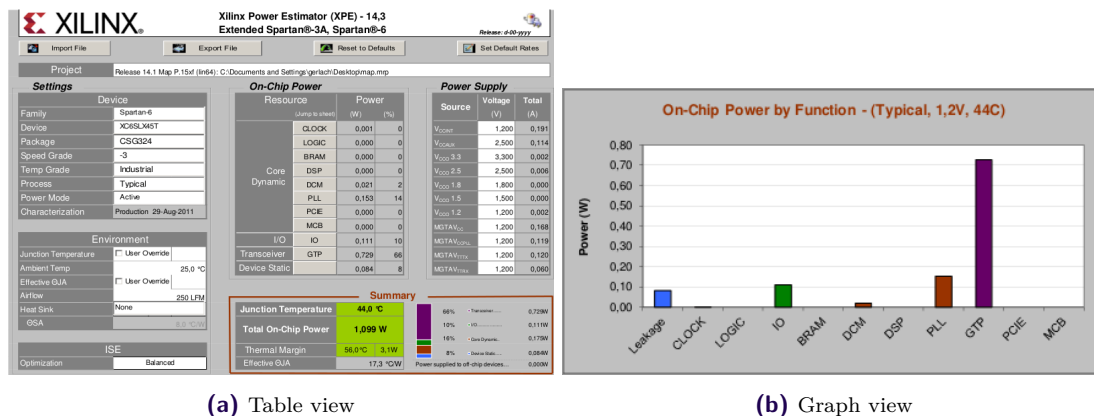
## Appendix



In this chapter, the results of both XPE and XPA are presented. Further, a translation table for the old and new IOB signal names is provided for a convenient understanding, as the schematic excerpts presented in this document are not updated to the present naming convention.

## A.1 FPGA Power Estimation

The estimation of XPE was used for the design of the IOB local power circuitries. The calculator tool is fed with an approximate utilization of the FPGA I/O banks, the expected clock domains, and the I/O standards being employed. On the other hand, XPA is provided with the netlist of the actual firmware logic, which is then analyzed in detail to calculate the power dissipation.



**Figure A.1:** Power estimation of present IOB firmware using Xilinx Power Estimator.

[illegible]

**Figure A.2:** Power analysis of present IOB firmware using Xilinx Power Analyzer.

## A.2 Signal Naming

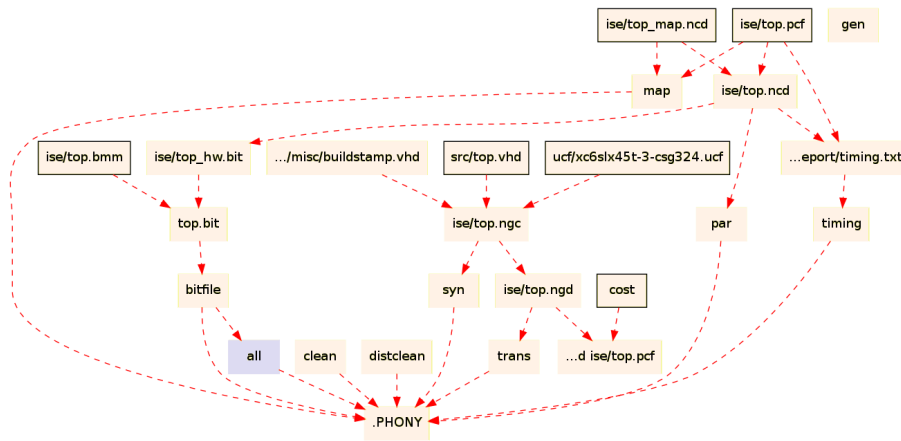
The signal naming has slightly changed from the beginning of IOB development to present DSSC status. The following table reflects the changes (bold), which mainly concern the ASIC interfaces.

Interface	Old	Signal name New	Description
ASIC control	START ABORT RESET	<b>XCLK</b> <b>XDATA</b> <b>XRESET</b>	FEE synchronization clock Detector-specific fast-control commands Global detector reset
ASIC data	CLK C_DO<1..16>	<b>ADCCLK</b> <b>ASIC_DO&lt;1..16&gt;</b>	ADC sampling clock ASIC data channels
High-speed transceivers	REFCLK101 MGT_TX<0..3>	<b>REFCLKGTP</b> MGT_TX<0..3>	GTP reference clock GTP transmission lines
Clear control	CLR CLRGATE CLRDIS	CLR CLRGATE <b>CLRDIS</b>	Sensor clear (drain) control signal Sensor clear gate control signal Sensor clear gate driver disable line (low-active)
PRB control	SR_CLK SR_DI SR_DO SR_RCLK SR_RST	SR_CLK SR_DI SR_DO SR_RCLK SR_RST	PRB shifreg clock PRB shiftreg serial data-in PRB shiftreg serial data-out PRB shiftreg latch enable PRB shiftreg reset
LMK control	CLKUWIRE DATAUWIRE LEUWIRE<1..4>	CLKUWIRE DATAUWIRE LEUWIRE<1..4>	Common LMK device clock Common LMK device serial data Separate LMK device latch enable
Sensor power	VGATE<1..2> VSOURCE<1..2> VSSS<1..2>	VGATE<1..2> VSOURCE<1..2> VSSS<1..2>	Sensor gate contact supply voltage Sensor source contact supply voltage Source / gate supply voltage reference
IOB power	IOBPOW<1..3>	IOBPOW<1..3>	IOB main power supply
IOB slow-control	SYNC<1..2> CNTR	<b>SYNCLK&lt;1..2&gt;</b> CNTR	Slow-control differential clock Slow-control bi-directional serial data
JTAG	TCK TDI TDO TMS	TCK TDI TDO TMS	Clock Serial data-in Serial data-out Test mode select

**Table A.1:** Signal naming of IOB interfaces. Changes are printed bold.

## B VHDL Firmware Modules

The configuration *bitstreams* for the FPGA are generated by *synthesizing* and *mapping* the VHDL code of the firmware to the utilized FPGA components using the *Xilinx ISE / EDK 14.1* application suite. The software tools are applied in a custom Makefile-based build flow which allows for a fully automated build system. The build flow has the advantage of periodically synthesizing the designs using the latest revision of the VHDL code, for example in a nightly build system. As a consequence, the validity of the code repository is continuously monitored. Although not implemented at the present time, a nightly build system is advisable for future development. Figure B.1 visualizes the Makefile dependencies of the build flow.



**Figure B.1:** Simplified visualization of Makefile-based build flow for the FPGA firmware.

This chapter provides some additional information about the IOB firmware modules. In particular, their VHDL code entity port declarations are given. Additionally, the waveforms of functional simulations are presented along with the individual SysConfig register maps.

### B.1 SysConfig

#### B.1.1 Entity Port Declaration

```
1  entity sysconf is
2    port (
3        clk           : in std_logic;
4        rst           : in std_logic;
5
6        we             : in std_logic;
```

```
7      addr          : in unsigned((C_SYSCONF_ADDR_WIDTH - 1) downto 0);
8      wrdata        : in std_logic_vector((C_SYSCONF_DATA_WIDTH - 1) downto 0);
9      rddata        : out std_logic_vector((C_SYSCONF_DATA_WIDTH - 1) downto 0);
10
11      lmk01010_en    : out std_logic;
12      lmk01010_valid : out std_logic;
13      lmk01010_dev_sel : out std_logic_vector(3 downto 0);
14      lmk01010_data   : out std_logic_vector(31 downto 0);
15      lmk01010_ack    : in std_logic;
16      lmk01010_busy   : in std_logic;
17
18      prb_en         : out std_logic;
19      prb_srrst      : out std_logic;
20      prb_gdps_dly    : out unsigned(31 downto 0);
21      prb_data10      : out std_logic_vector(31 downto 0);
22      prb_data32      : out std_logic_vector(31 downto 0);
23      prb_ack         : in std_logic;
24      prb_busy       : in std_logic;
25
26      fetdrv_en       : out std_logic;
27      fetdrv_dvgate_on : out unsigned(31 downto 0);
28      fetdrv_dvsource_on : out unsigned(31 downto 0);
29      fetdrv_dvsss_on  : out unsigned(31 downto 0);
30      fetdrv_dvgate_off : out unsigned(31 downto 0);
31      fetdrv_dvsource_off : out unsigned(31 downto 0);
32      fetdrv_dvsss_off : out unsigned(31 downto 0);
33
34      clr_en          : out std_logic;
35      clr_preclr_dly   : out unsigned(31 downto 0);
36      clr_clron_ofs    : out unsigned(1 downto 0);
37      clr_clroff_ofs   : out unsigned(1 downto 0);
38      clr_clrgateon_ofs : out unsigned(1 downto 0);
39      clr_clrgateoff_ofs : out unsigned(1 downto 0);
40      clr_clr_period   : out unsigned(31 downto 0);
41      clr_clr_duty      : out unsigned(31 downto 0);
42      clr_clrdis_duty   : out unsigned(31 downto 0);
43      clr_ack          : in std_logic;
44      clr_busy         : in std_logic;
45  );
46  end entity sysconf;
```



## B.2 SysCtrl

### B.2.1 Entity Port Declaration

```

1  entity sysctrl is
2      port (
3          clk                : in  std_logic;
4          rst                : in  std_logic;
5          cmdclk             : in  std_logic;
6
7          cntr_in            : in  std_logic;
8          cntr_out           : out std_logic;
9          oe                 : out std_logic;
10
11         debug              : out std_logic_vector(31 downto 0);
12
13         sysconf_we         : out std_logic;
14         sysconf_addr       : out unsigned(15 downto 0);
15         sysconf_datain     : in  std_logic_vector(31 downto 0);
16         sysconf_dataout    : out std_logic_vector(31 downto 0);
17
18         ack                : out std_logic;
19         busy               : out std_logic;
20     );
21 end entity sysctrl;

```

### B.2.2 Simulation

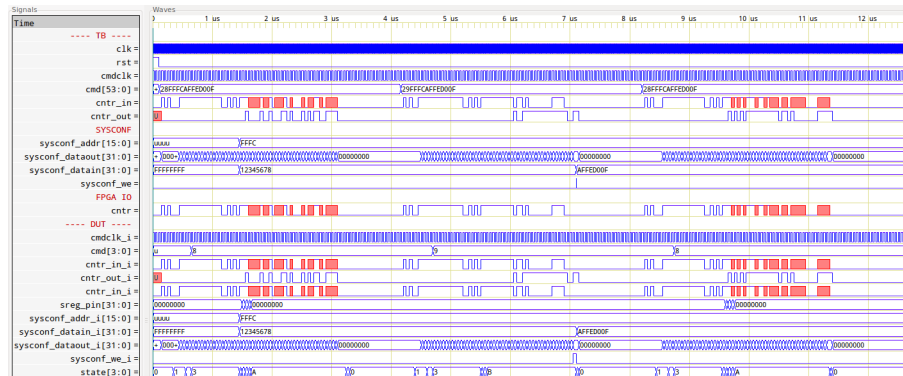


Figure B.2: Waveform of SysCtrl simulation. Simulated with GHDL.

## B.3 LmkCtrl

### B.3.1 Entity Port Declaration

```

1  entity lmk01010_ctrl is
2      port (
3          refclk      : in std_logic;
4          clk         : in std_logic;
5          rst         : in std_logic;
6
7          en          : in std_logic;
8          valid       : in std_logic;
9          dev_sel     : in std_logic_vector((C_NUM_DEV - 1) downto 0);
10
11         reg_addr    : in std_logic_vector(3 downto 0);
12         reg_data    : in std_logic_vector(27 downto 0);
13
14         clk_uwire   : out std_logic;
15         data_uwire  : out std_logic;
16         le_uwire    : out std_logic_vector((C_NUM_DEV - 1) downto 0);
17
18         ack         : out std_logic;
19         busy        : out std_logic;
20     );
21 end entity lmk01010_ctrl;

```

### B.3.2 SysConfig Register Map

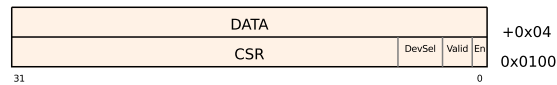


Figure B.3: SysConfig register map for the LmkCtrl module.

### B.3.3 Simulation

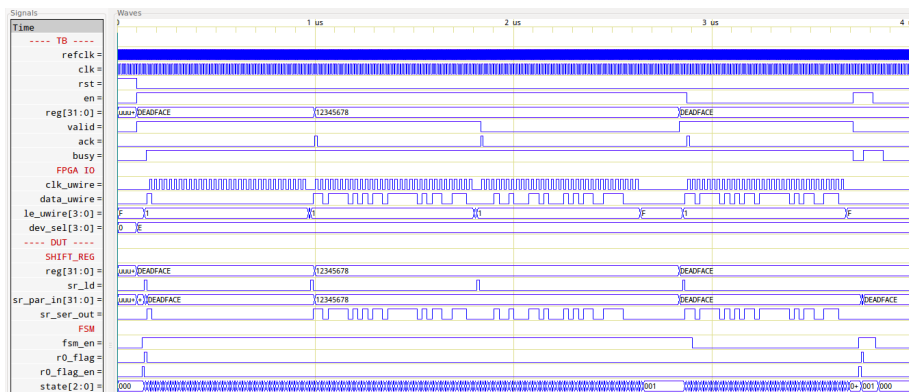


Figure B.4: Waveform of LmkCtrl simulation. Simulated with GHDL.

## B.4 PrbCtrl

### B.4.1 Entity Port Declaration

```

1  entity prb_ctrl_sw is
2      port (
3          clk          : in  std_logic;
4          rst          : in  std_logic;
5
6          datain       : in  std_logic_vector(63 downto 0);
7
8          init         : out  std_logic;
9          valid        : out  std_logic;
10         dataout       : out  std_logic_vector(((C_NUM_PRB * 16) - 1) downto 0);
11         srctrl_srrst  : out  std_logic;
12         srctrl_ack    : in  std_logic;
13         srctrl_busy   : in  std_logic;
14
15         en           : in  std_logic;
16         srrst        : in  std_logic;
17         gdps_dly     : in  unsigned(31 downto 0);
18
19         ack          : out  std_logic;
20         busy         : out  std_logic
21     );
22 end entity prb_ctrl_sw;

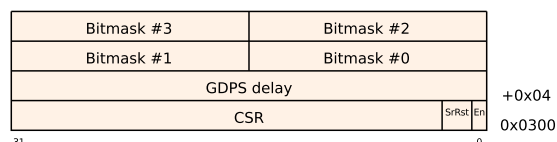
```

```

1  entity prb_sr_ctrl is
2      port (
3          refclk       : in  std_logic;
4          clk          : in  std_logic;
5          rst          : in  std_logic;
6
7          en           : in  std_logic;
8          rst_sr       : in  std_logic;
9          valid        : in  std_logic;
10         data         : in  std_logic_vector(((C_NUM_PRB * 16) - 1) downto 0);
11
12         sr_clk       : out  std_logic;
13         sr_rst       : out  std_logic;
14         sr_rclk      : out  std_logic;
15         sr_di        : out  std_logic;
16         sr_do        : in  std_logic;
17
18         ack          : out  std_logic;
19         busy         : out  std_logic
20     );
21 end entity prb_sr_ctrl;

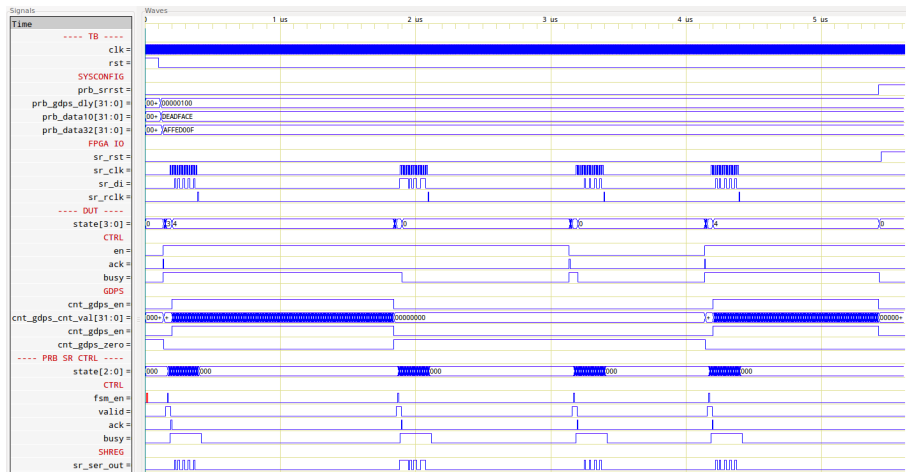
```

### B.4.2 SysConfig Registers

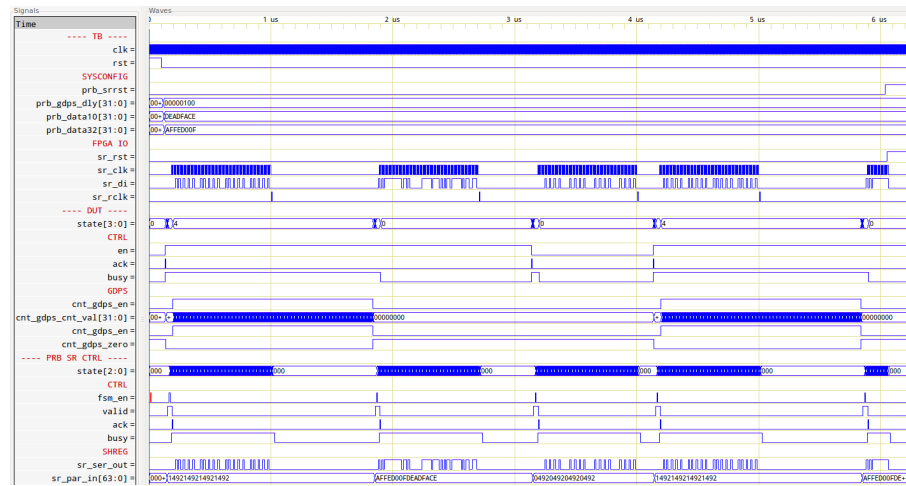


**Figure B.5:** SysConfig register map for the PrbCtrl module.

### B.4.3 Simulation



(a) 16-bit (single PRB)



(b) 64-bit (four PRBs)

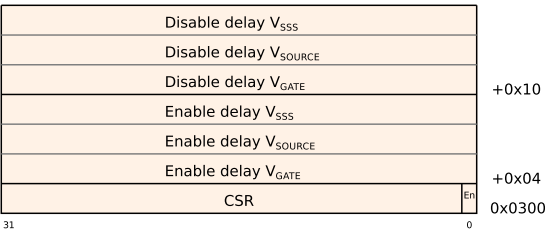
**Figure B.6:** Waveform of PrbCtrl simulation. Simulated with GHDL.

## B.5 FetCtrl

### B.5.1 Entity Port Declaration

```
1  entity fetdrv_ctrl is
2      port (
3          clk          : in  std_logic;
4          rst          : in  std_logic;
5
6          en           : in  std_logic;
7          dvgate_on    : in  unsigned(31 downto 0);
8          dvsource_on  : in  unsigned(31 downto 0);
9          dvsss_on     : in  unsigned(31 downto 0);
10         dvgate_off   : in  unsigned(31 downto 0);
11         dvsource_off : in  unsigned(31 downto 0);
12         dvsss_off    : in  unsigned(31 downto 0);
13
14         vgate_ctrl   : out std_logic;
15         vsource_ctrl : out std_logic;
16         vsss_ctrl    : out std_logic;
17     );
18 end entity fetdrv_ctrl;
```

### B.5.2 SysConfig Registers



**Figure B.7:** SysConfig register map for the FetCtrl module.

### B.5.3 Simulation



**Figure B.8:** Waveform of FetCtrl simulation. Simulated with GHDL.

## B.6 ClrCtrl

### B.6.1 Entity Port Declaration

```
1  entity clr_ctrl_oserdes2 is
2      port (
3          rst_async      : in std_logic;
4          ioclk          : in std_logic;
5          gclk           : in std_logic;
6          serdesstrobe   : in std_logic;
7          rst            : in std_logic;
8
9          en             : in std_logic;
10
11         preclr_dly      : in unsigned(31 downto 0);
12         clron_ofs       : in unsigned(1 downto 0);
13         clroff_ofs      : in unsigned(1 downto 0);
14         clrgateon_ofs   : in unsigned(1 downto 0);
15         clrgateoff_ofs  : in unsigned(1 downto 0);
16         clr_period      : in unsigned(31 downto 0);
17         clr_duty        : in unsigned(31 downto 0);
18         clrdys_duty     : in unsigned(31 downto 0);
19
20         clr             : out std_logic;
21         clrdys          : out std_logic;
22         clrgate         : out std_logic;
23
24         debug           : out std_logic_vector(31 downto 0);
25
26         ack             : out std_logic;
27         busy            : out std_logic;
28     );
29 end entity;
```

### B.6.2 SysConfig Registers

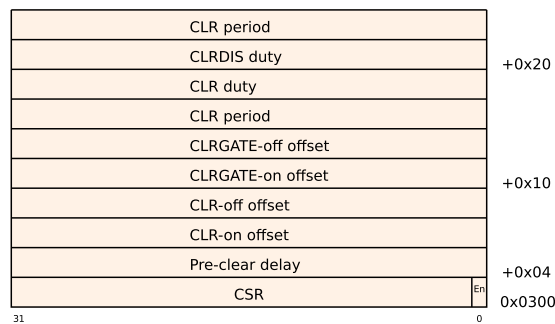
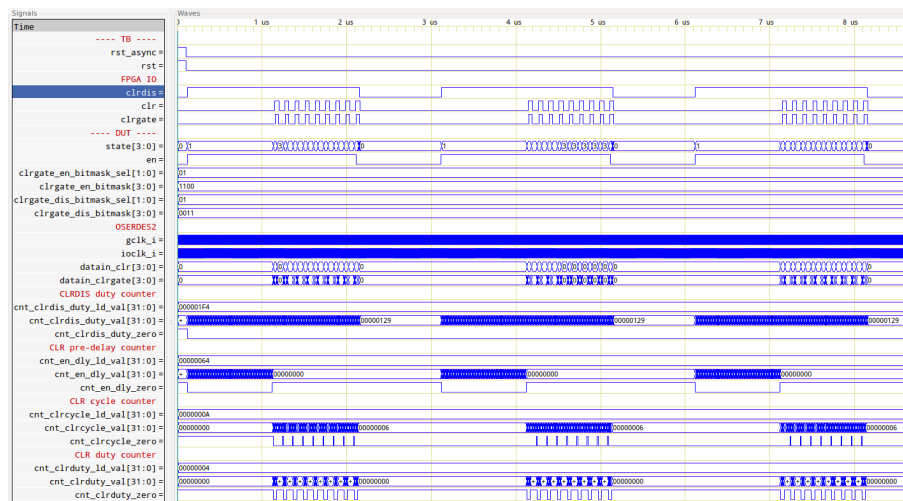
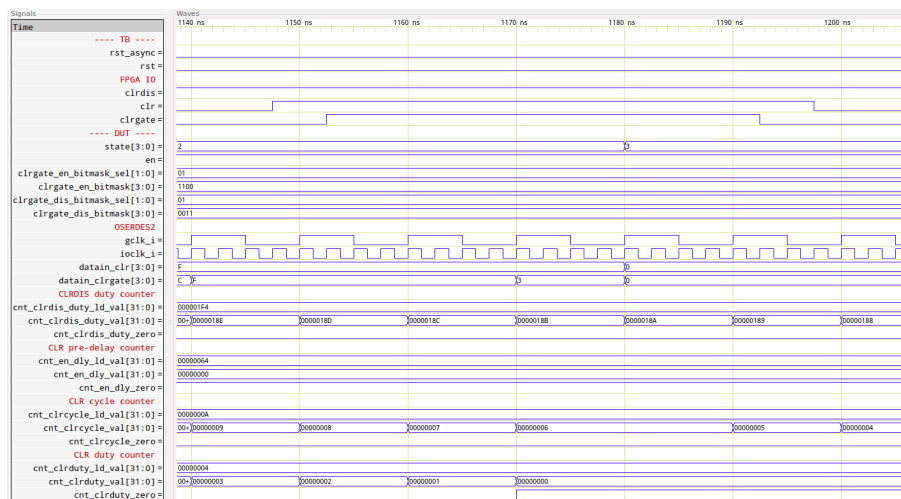


Figure B.9: SysConfig register map for the ClrCtrl module.

## B.6.3 Simulation



(a) Sequencing



(b) CLR-CLRGATE relation

Figure B.10: Waveform of ClrCtrl simulation. Simulated with GHDL.



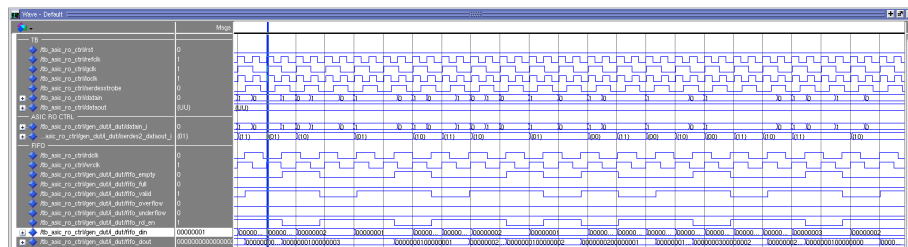
## B.7 AsicRoCtrl

### B.7.1 Entity Port Declaration

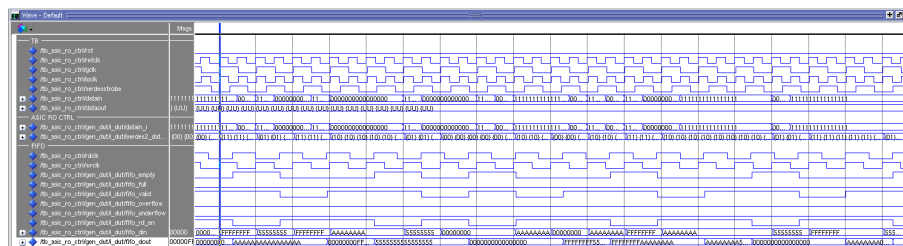
```

1  entity asic_ro_ctrl is
2      port (
3          rd_en          : in std_logic;
4          wr_en          : in std_logic;
5
6          rd_clk         : in std_logic;
7          wr_clk         : in std_logic;
8          gclk           : in std_logic;
9          ioclk          : in std_logic;
10         rst             : in std_logic;
11         rst_async       : in std_logic;
12
13         serdesstrobe    : in std_logic;
14
15         empty           : out std_logic;
16         valid           : out std_logic;
17
18         datain          : in std_logic_vector((C_DIN_WIDTH - 1) downto 0);
19         dataout         : out std_logic_vector((C_DOUT_WIDTH - 1) downto 0);
20         debug           : out std_logic_vector(63 downto 0)
21     );
22 end entity;
```

### B.7.2 Simulation



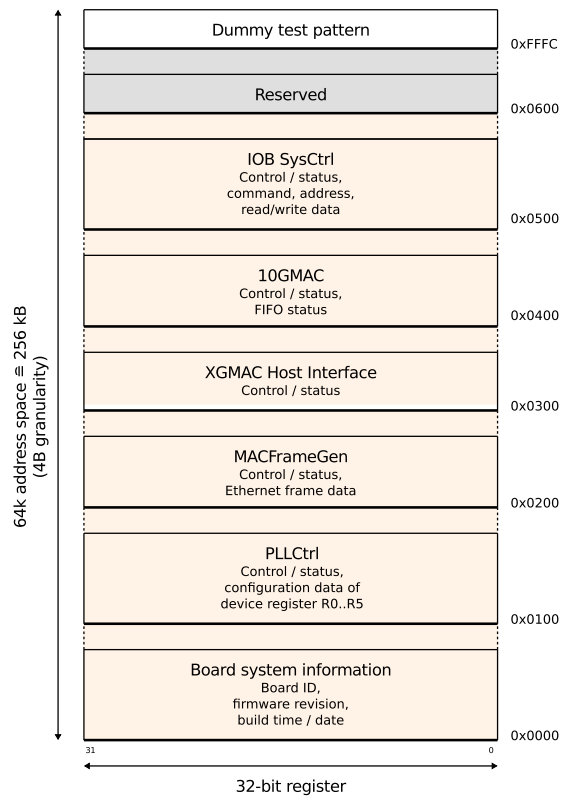
(a) A single ASIC



(b) Sixteen ASICs

Figure B.11: Waveform of AsicRoCtrl simulation. Simulated with ModelSim.

## B.8 MPRACE-2 SysConfig Map



**Figure B.12:** Register map of MPRACE-2 SysConfig.





# Acronyms

<b>AC</b>	Alternating Current
<b>ADC</b>	Analog Digital Converter
<b>AGIPD</b>	Adaptive Gain Integrating Pixel Detector
<b>ALICE</b>	A Large Ion Collider Experiment
<b>ARM</b>	Advanced RISC Machine
<b>ASIC</b>	Application Specific Integrated Circuit
<b>AsicRoCtrl</b>	ASIC Readout Controller
<b>ATB</b>	ASIC Test Board
<b>ATCA</b>	Advanced Telecommunications Computing Architecture
<b>ATLAS</b>	A Toroidal LHC Apparatus
<b>BER</b>	Bit Error Rate
<b>BJT</b>	Bipolar Junction Transistor
<b>C&amp;C</b>	Clock And Control
<b>CERN</b>	Conseil Européen pour la Recherche Nucléaire
<b>ClkGen</b>	Clock Generator
<b>ClrCtrl</b>	Clear Controller
<b>CML</b>	Current Mode Logic
<b>CMS</b>	Compact Muon Solenoid
<b>CPU</b>	Central Processing Unit
<b>CSR</b>	Control Status Register
<b>DAQ</b>	Data Acquisition
<b>DC</b>	Direct Current
<b>DDR2</b>	Double Data Rate
<b>DDR3</b>	Double Data Rate
<b>DEMUX</b>	De-Multiplexer
<b>DEPFET</b>	Depleted P-channel Field Effect Transistor

<b>DESY</b>	Deutsches Elektron Synchrotron
<b>DMA</b>	Direct Memory Access
<b>DSSC</b>	DEPFET Sensor with Signal Compression
<b>FEE</b>	Front End Electronics
<b>FEL</b>	Free Electron Laser
<b>FEM</b>	Front-End Module
<b>FET</b>	Field Effect Transistor
<b>FetCtrl</b>	FET Controller
<b>FIFO</b>	First-In First-Out
<b>FMC</b>	FPGA Mezzanine Card
<b>FPGA</b>	Field Programmable Gate Array
<b>FSM</b>	Finite State Machine
<b>GbE</b>	Gigabit Ethernet
<b>GPU</b>	Graphics Processing Unit
<b>GTP</b>	Gigabit Transceiver at low Power
<b>GTX</b>	Extended Gigabit Transceiver
<b>HLL</b>	Halbleiter Labor
<b>HLT</b>	High Level Trigger
<b>I/O</b>	Input/Output
<b>ID</b>	Identification
<b>IOB</b>	I/O Board
<b>IOBUF</b>	I/O Buffer
<b>IODELAY</b>	Input/Output Delay
<b>IOSERDES</b>	Input/Output Serializer/Deserializer
<b>IP</b>	Internet Protocol
<b>ISERDES</b>	Input Serializer/Deserializer
<b>Laser</b>	Light Amplification by Stimulated Emission of Radiation
<b>LC</b>	Lucent Connector
<b>LED</b>	Light-Emitting Diode
<b>LHC</b>	Large Hadron Collider

<b>LHCb</b>	Large Hadron Collider beauty
<b>LmkCtrl</b>	LMK01010 Controller
<b>LPD</b>	Large Pixel Detector
<b>LSBit</b>	Least Significant Bit
<b>LVCMOS</b>	Low-Voltage Complementary Metal-Oxide Semiconductor
<b>LVDS</b>	Low-Voltage Differential Signaling
<b>MAC</b>	Media Access Control
<b>MGT</b>	Multi-Gigabit Transceiver
<b>MIB</b>	Module Interconnection Board
<b>MOS</b>	Metal Oxide Semiconductor
<b>MOSFET</b>	Metal Oxide Semiconductor Field Effect Transistor
<b>MPE</b>	Max-Planck Institut für extraterrestrische Physik
<b>MPO</b>	Multi-fiber Push-On
<b>MPRACE-2</b>	Massive Parallel Readout Accelerator v.2
<b>MSBit</b>	Most Significant Bit
<b>MTCA</b>	Micro Telecommunications Computing Architecture
<b>MUX</b>	Multiplexer
<b>OS</b>	Operating System
<b>OSERDES</b>	Output Serializer/Deserializer
<b>PC</b>	Personal Computer
<b>PCB</b>	Printed Circuit Board
<b>PCIe</b>	Peripheral Component Interconnect Express
<b>PHY</b>	Physical Layer (of the OSI model)
<b>PLL</b>	Phase-Locked Loop
<b>PPT</b>	Patch Panel Transceiver
<b>PRB</b>	Power Regulator Board
<b>PrbCtrl</b>	Power Regulator Board Controller
<b>PrbSRCtrl</b>	Power Regulator Board Shift Register Controller
<b>QSFP+</b>	Quad Small Form-Factor Pluggable (Transceiver)
<b>RAM</b>	Random-Access Memory

<b>RC</b>	Resistance / Capacitance
<b>RX</b>	Receiver
<b>SASE</b>	Self-amplified Spontaneous Emission
<b>SDRAM</b>	Synchronous Dynamic Random Access Memory
<b>SerDes</b>	Serializer / De-serializer
<b>SFP</b>	Small Form-factor Pluggable (Transceiver)
<b>SFP+</b>	Extended SFP
<b>SOF</b>	Start of Frame
<b>SRAM</b>	Synchronous Random Access Memory
<b>STFC</b>	Science and Technology Facilities Council
<b>SysConfig</b>	System Configuration
<b>SysCtrl</b>	System Controller
<b>TB</b>	Train Builder
<b>TCP</b>	Transmission Control Protocol
<b>TX</b>	Transmitter
<b>UART</b>	Universal Asynchronous Receiver / Transmitter
<b>UDP</b>	User Datagram Protocol
<b>VHDL</b>	VHSIC Hardware Description Language
<b>VHSIC</b>	Very High-Speed Integrated Circuits
<b>XAUI</b>	X (Ten) Attachment Unit Interface
<b>XFEL</b>	X-ray Free Electron Laser
<b>XO</b>	Crystal Oscillator
<b>XPA</b>	Xilinx Power Analyzer
<b>XPE</b>	Xilinx Power Estimator
<b>ZITI</b>	Zentrales Institut für Technische Informatik (Central Institute for Computer Engineering)



# List of Figures

1.1	Fields of application of the European XFEL . . . . .	7
2.1	Principle of an FEL . . . . .	12
2.2	Synchrotron radiation . . . . .	13
2.3	Resonance condition in an undulator . . . . .	13
2.4	SASE principle and micro-bunching . . . . .	14
2.5	The construction area of the European XFEL . . . . .	15
2.6	Beamlines of the European XFEL . . . . .	15
2.7	Electron bunch timing of the European XFEL . . . . .	18
3.1	Schematic view of a DEPFET . . . . .	24
3.2	Cross section and characteristics of standard DEPFET and DSSC . . . . .	25
3.3	Schematic view of the DSSC detector head . . . . .	25
3.4	Concept of the DSSC detector . . . . .	27
3.5	Illustration of DSSC detector layout . . . . .	27
3.6	Back-end DAQ of the European XFEL, shown for one 2d detector. . . . .	29
3.7	Train Builder board setup for a megapixel detector. . . . .	30
3.8	Train building with crosspoint for a quarter megapixel detector. . . . .	31
3.9	Concept of the C&C system . . . . .	31
3.10	Pinout of the C&C fast signal interface to the detector FEE . . . . .	32
3.11	Examples of fixed and variable latency veto protocol . . . . .	34
3.12	Block diagram of the DSSC readout ASIC . . . . .	35
4.1	Concept of the AGIPD device and its DAQ electronics . . . . .	40
4.2	Concept of the front-end DAQ system of the LPD . . . . .	41
5.1	Concept of the DSSC front-end DAQ . . . . .	44
5.2	State transition graph of PPT timing FSM . . . . .	46
5.3	Concept and example of the DSSC ASIC veto mechanism . . . . .	48
5.4	Data path of the DSSC DAQ . . . . .	49
5.5	Layout and connectivity of the PPT prototype . . . . .	52
5.6	DSSC patch panel with location of PPT cards . . . . .	53
5.7	MPO-to-LC breakout cable by Molex . . . . .	53
5.8	Sketch of IOB layout . . . . .	54
5.9	Connectivity of IOB FPGA . . . . .	55
5.10	DAQ connectivity of a DSSC quadrant . . . . .	57
6.1	Top and bottom view of IOB prototype . . . . .	59
6.2	Length matching of differential traces . . . . .	62
6.3	I/O bank utilization of IOB Spartan-6 FPGA . . . . .	65
6.4	Top and bottom side of the IOB-rev0.1 blank . . . . .	66

6.5	Layer stack-up of the IOB PCB . . . . .	67
6.6	Illustration of differential trace parameters . . . . .	69
6.7	Circuitries for IOB local power supply . . . . .	72
6.8	Switching circuitries for $V_{SOURCE}$ and $V_{GATE}$ . . . . .	73
6.9	Switching circuitry for $V_{SSS}$ . . . . .	74
6.10	Schematics of control circuits for sensor clear signal gate drivers . . . . .	75
6.11	Three-dimensional model of IOB-rev1.0 . . . . .	76
7.1	Block diagram of IOB firmware . . . . .	77
7.2	Schematic of ClkGen module . . . . .	79
7.3	Possible transition graph for IOB master FSM . . . . .	80
7.4	Structure of the SysConfig register bank . . . . .	81
7.5	Schematic of SysConfig logic . . . . .	81
7.6	Schematic of SysCtrl logic . . . . .	83
7.7	State transition graph of SysCtrl FSM . . . . .	83
7.8	Preliminary timing of ASIC power switching . . . . .	85
7.9	Schematic of PrbCtrl module . . . . .	86
7.10	State transition graph of PrbCtrl FSM . . . . .	86
7.11	Schematic of PrbSRCtrl module . . . . .	87
7.12	State transition graph of PrbSRCtrl FSM . . . . .	87
7.13	Schematic of LmkCtrl logic . . . . .	88
7.14	State transition graph of LmkCtrl FSM . . . . .	89
7.15	Preliminary timing of sensor clear signals . . . . .	90
7.16	Relation between sensor clear and clear gate signal . . . . .	90
7.17	Schematic graph of ClrCtrl logic . . . . .	91
7.18	State transition graph of ClrCtrl FSM . . . . .	91
7.19	Preliminary timing of sensor power switching . . . . .	92
7.20	Schematic of FetCtrl logic . . . . .	93
7.21	Concept of ASIC readout . . . . .	94
7.22	Schematic of ISERDES data de-serialization module . . . . .	94
7.23	ASIC data serialization . . . . .	95
7.24	Layout of ASIC readout FIFO . . . . .	96
7.25	Block diagram of Aurora TX core . . . . .	96
7.26	IOB test environment . . . . .	98
7.27	Photograph and block diagram of the MPRACE-2 board . . . . .	99
7.28	10GbE / PLL mezzanine for the MPRACE-2 . . . . .	99
7.29	Block diagram of MPRACE-2 firmware . . . . .	100
7.30	Screenshot of MPRACE-2 software . . . . .	101
8.1	ChipScope capture of SysCtrl read and write transactions . . . . .	104
8.2	Oscilloscope capture of SysCtrl read and write transactions . . . . .	105
8.3	Signal measurement of ADC clock . . . . .	105
8.4	Signal measurement of ASIC command clock / data . . . . .	106
8.5	Signal measurement on ASIC data lane of IOB . . . . .	106
8.6	ChipScope waveform of AsicRoCtrl test . . . . .	106
8.7	Signal measurement of GTPs and Aurora . . . . .	107
8.8	ChipScope recording of MPRACE-2 Aurora RX . . . . .	108
8.9	BER test of IOB GTPs . . . . .	108
8.10	ChipScope capture of PrbCtrl enabling and disabling programming sequence . . . . .	109

8.11	Signal measurement of PrbCtrl programming sequences . . . . .	110
8.12	Test measurement of ASIC power cycling . . . . .	111
8.13	ChipScope recording of LmkCtrl programming sequence . . . . .	112
8.14	Oscilloscope capture of LmkCtrl programming sequence . . . . .	112
8.15	Signal measurement of Main Board clock buffer delay adjustment . . . . .	113
8.16	ChipScope capture of FetCtrl enabling and disabling sequence . . . . .	114
8.17	Gating of power net $V_{SOURCE}$ . . . . .	115
8.18	Gate voltage of the controlling p-channel MOSFET in the $V_{GATE}$ circuitry . . . . .	116
8.19	Signal measurement of ClrCtrl control signals . . . . .	117
8.20	Clear signal sequencing . . . . .	118
9.1	Proposal for the DAQ test setup of a single module . . . . .	123
A.1	FPGA power estimation with XPE . . . . .	127
A.2	FPGA power analysis with XPA . . . . .	127
B.1	Makefile-based build flow . . . . .	129
B.2	Waveform of SysCtrl simulation . . . . .	131
B.3	SysConfig register map for LmkCtrl . . . . .	132
B.4	Waveform of LmkCtrl simulation . . . . .	132
B.5	SysConfig register map for PrbCtrl . . . . .	133
B.6	Waveform of PrbCtrl simulation . . . . .	134
B.7	SysConfig register map for FetCtrl . . . . .	135
B.8	Waveform output of FetCtrl simulation . . . . .	136
B.9	SysConfig register map for ClrCtrl . . . . .	137
B.10	Waveform of ClrCtrl simulation . . . . .	138
B.11	Waveform of AsicRoCtrl simulation . . . . .	139
B.12	Register map of MPRACE-2 SysConfig . . . . .	140



## List of Tables

2.1	Key characteristics of the European XFEL . . . . .	16
2.2	Characteristics of the SASE systems and the undulators . . . . .	16
2.3	Applications of XFEL experiments . . . . .	18
3.1	DSSC detector key characteristics . . . . .	26
3.2	Command protocol of the Clock & Control system . . . . .	32
3.3	Veto protocol of the Clock & Control system . . . . .	33
4.1	Payload data rates inside the AGIPD DAQ system. . . . .	40
5.1	Summary of the DSSC ASIC readout commands . . . . .	47
5.2	Interface bandwidths of the DSSC DAQ . . . . .	50
5.3	Comparison of different FPGAs from Altera, Lattice, and Xilinx . . . . .	51
6.1	I/O bank usage of IOB FPGA . . . . .	65
6.2	Key parameters of the IOB PCB . . . . .	66
6.3	Stack-up and via properties of the IOB PCB . . . . .	68
6.4	Split ground planes applied on the IOB prototype . . . . .	68
6.5	Impedances of differential traces on the IOB . . . . .	69
6.6	Summary of trace length tolerances of the differential signals . . . . .	69
6.7	Bypass capacitors for the IOB . . . . .	70
6.8	Estimation of IOB power dissipation . . . . .	71
7.1	SysCtrl communication protocol . . . . .	82
7.2	Bit assignment of PRB shift registers . . . . .	85
7.3	Supported commands of MPRACE-2 software. . . . .	101
A.1	Signal naming of IOB interfaces . . . . .	128



# Bibliography

- [1] European XFEL GmbH. *The European XFEL*. URL: <http://www.xfel.eu/>.
- [2] E. Krinsky, M. L. Perlman, and R. E. Watson. “Characteristics of Synchrotron Radiation and of its Sources”. In: *Handbook on Synchrotron Radiation*. Ed. by E. E. Koch. 1983, pp. 65–171.
- [3] K. J. Kim. “Characteristics of Synchrotron Radiation”. In: *AIP Conference Proceedings* 185 (1987), pp. 565–632.
- [4] T. Nakazato et al. “Observation of Coherent Synchrotron Radiation”. In: *Physical Review Letters* 63.12 (1989), pp. 1245–1248.
- [5] E. B. Blum, U. Happek, and A. J. Sievers. “Observation of Coherent Synchrotron Radiation at the Cornell Linac”. In: *Nuclear Instruments and Methods* A307.2–3 (1991), pp. 568–576.
- [6] European XFEL GmbH. *How Does It Work?* URL: [http://www.xfel.eu/overview/how\\_does\\_it\\_work/](http://www.xfel.eu/overview/how_does_it_work/).
- [7] DESY HASYLAB. *The XFEL Principle*. 14–15. URL: [https://hasylab.desy.de/e70/e6129/e4242/e4370/felbasics\\_eng.pdf](https://hasylab.desy.de/e70/e6129/e4242/e4370/felbasics_eng.pdf).
- [8] S. Krinsky. *The Physics and Properties of Free-Electron Lasers*. 2002.
- [9] European XFEL GmbH. *The European XFEL: Facts and Figures*. URL: [http://www.xfel.eu/overview/facts\\_and\\_figures/](http://www.xfel.eu/overview/facts_and_figures/).
- [10] European XFEL GmbH. *Beamlines*. URL: <http://www.xfel.eu/beamlines/>.
- [11] Th. Tschentscher. *Layout of the X-Ray Systems at the European XFEL*. Tech. rep. 001. 2011.
- [12] European XFEL GmbH. *Instruments*. URL: <http://www.xfel.eu/research/instruments/>.
- [13] H. Graafsma. “Requirements for and Development of 2-dimensional X-ray detectors for the European X-ray Free Electron Laser in Hamburg”. In: *Journal of Instrumentation* 4 (2009), pp. 29–33.
- [14] A. Mozzanica et al. “A Single Photon Resolution Integrating Chip for Microstrip Detectors”. In: *Nuclear Instruments and Methods in Physics* 633.1 (2011), pp. 29–33.
- [15] D. Greiffenberg. “The AGIPD Detector for the European XFEL”. In: *Journal of Instrumentation* 7.1 (2012).
- [16] *AGIPD - Adaptive Gain Integrating Pixel Detector*. URL: [http://hasylab.desy.de/instrumentation/detectors/projects/agipd/index\\_eng.html](http://hasylab.desy.de/instrumentation/detectors/projects/agipd/index_eng.html).
- [17] M. Porro. “Development of the DEPFET Sensor with Signal Compression: A Large Format X-ray Imager with Mega-frame Readout Capability for the European XFEL”. In: *Nuclear Science Symposium and Medical Imaging Conference*. 2011, pp. 1424–1434.
- [18] European XFEL GmbH. *Data Handling*. URL: [http://www.xfel.eu/research/data\\_handling/](http://www.xfel.eu/research/data_handling/).
- [19] J. Kemmer and G. Lutz. “New Detector Concepts”. In: *Nuclear Instruments and Methods in Physics* 253.3 (1987), pp. 365–377.

- [20] J. Kemmer et al. “Experimental Confirmation of a New Semiconductor Detector Principle”. In: *Nuclear Instruments and Methods in Physics* 288.1 (1990), pp. 92–98.
- [21] G. Lutz et al. “DEPFET Sensor with Intrinsic Signal Compression Developed for Use at the XFEL Free Electron Laser Radiation Source”. In: *Nuclear Instruments and Methods in Physics* 624.2 (2010), pp. 528–532.
- [22] P. Lechner et al. “DEPFET Active Pixel Sensor with Non-linear Amplification”. In: Nuclear Science Symposium and Medical Imaging Conference. 2011, pp. 563–568.
- [23] Max-Planck-Institut Halbleiterlabor. *The Depleted P-channel Field Effect Transistor (DEPFET)*. URL: <http://twiki.hll.mpg.de/twiki/bin/view/DEPFET/DepfetPrinciple>.
- [24] J. Coughlan et al. “The Train Builder Data Acquisition System for the European XFEL”. In: *Journal of Instrumentation* 6.11 (2011).
- [25] E. Motuk et al. “Design and Development of Electronics for the EuXFEL Clock and Control System”. In: *Journal of Instrumentation* 7.1 (2012).
- [26] P. Goettlicher, I. Sheviakov, and M. Zimmer. “10G-Ethernet Prototyping for 2-D X-Ray Detectors at the XFEL”. In: IEEE Real Time Conference. 2009, pp. 434–437.
- [27] M. Postranecky, M. Warren, and D. Wilson. *XFEL 2D Pixel Clock and Control System*. 2010.
- [28] M. Postranecky, M. Warren, and D. Wilson. *Clock and Control Fast Signal Specification*. 2010.
- [29] E. Motuk et al. “Experiences with the MTCA.4 Solution for the EuXFEL Clock and Control System”. In: IEEE Real Time Conference. 2012. Chap. PS4-9.
- [30] P. Fischer et al. “Pixel Readout ASIC with per Pixel Digitization and Digital Storage for the DSSC Detector at XFEL”. In: 2010, pp. 336–341.
- [31] S. Facchinetti et al. “Fast, Low-noise, Low-power Electronics for the Analog Readout of Non-Linear DEPFET Pixels”. In: 2011, pp. 1846–1851.
- [32] K. Hansen et al. “Pixel-level 8-bit 5-MS/s Wilkinson-type Digitizer for the DSSC X-ray Imager: Concept Study”. In: *Nuclear Instruments and Methods in Physics* 629.1 (2011), pp. 269–276.
- [33] M. Manghisoni et al. “High Accuracy Injection Circuit for Pixel-level Calibration of Readout Electronics”. In: 2010, pp. 1312–1318.
- [34] A. Kugel. “The ATLAS ROBIN – A High-Performance Data-Acquisition Module”. Dissertation. University of Mannheim, 2009.
- [35] D. Esperante et al. “LHCb Silicon Tracker DAQ and DCS Online Systems”. In: 2009, pp. 259–266.
- [36] J. de Cuveland. “A Track Reconstructing Low-latency Trigger Processor for High-energy Physics”. Dissertation. University of Heidelberg, 2009.
- [37] P. Goettlicher. “The Electronics in the Detector Head of the AGIPD Detector — A 1MPixel, 5MHz Camera for the European XFE”. In: Nuclear Science Symposium and Medical Imaging Conference. 2009, pp. 1811–1816.
- [38] P. Goettlicher et al. “High-speed Cameras for X-rays: AGIPD and Others”. In: Topical Workshop on Electronics for Particle Physics. 2012.
- [39] J. Coughlan et al. “The Data Acquisition Card for the Large Pixel Detector at the European XFEL”. In: *Journal of Instrumentation* 6 (2011).



- 
- [40] Heidelberg University. *MPRACE-2*. 2007. URL: <http://li5.ziti.uni-heidelberg.de/research/fpga/mprace2/>.
  - [41] F. Erdinger, P. Fischer, and J. Soldat. *DSSC MM3 Matrix Chip Manual*. 2012.
  - [42] G. A. Marcus Martinez. “Acceleration of Astrophysical Simulations with Special Hardware”. Dissertation. University of Heidelberg, 2011.
  - [43] URL: <http://www.molex.com>.
  - [44] Analog Devices. *ADF4351: Wideband Synthesizer with Integrated VCO*. 2012. URL: <http://www.analog.com/en/rfif-components/pll-synthesizersvcos/adf4351/products/product.html>.
  - [45] Xilinx. *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*. 2011. URL: [http://www.xilinx.com/support/documentation/data\\_sheets/ds162.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds162.pdf).
  - [46] H. Johnson and M Graham. *High-Speed Digital Design - A Handbook of Black Magic*. 1993.
  - [47] H. Johnson and M Graham. *High-Speed Signal Propagation - Advanced Black Magic*. 2003.
  - [48] Lee W. Ritchey. *Right the First Time: A Practical Handbook on High-Speed PCB and System Design*. Vol. 1. 2003.
  - [49] Xilinx. *Spartan-6 FPGA User Guide: Configuration*. 2012. URL: [http://www.xilinx.com/support/documentation/user\\_guides/ug380.pdf](http://www.xilinx.com/support/documentation/user_guides/ug380.pdf).
  - [50] Xilinx. *Spartan-6 FPGA Documentation*. 2009–2012. URL: [http://www.xilinx.com/support/documentation/user\\_guides/](http://www.xilinx.com/support/documentation/user_guides/).
  - [51] Xilinx. *Spartan-6 FPGA User Guide: PCB Design and Pin Planning Guide*. 2012. URL: [http://www.xilinx.com/support/documentation/user\\_guides/ug393.pdf](http://www.xilinx.com/support/documentation/user_guides/ug393.pdf).
  - [52] Xilinx. *Spartan-6 FPGA User Guide: GTP Transceivers*. 2010. URL: [http://www.xilinx.com/support/documentation/user\\_guides/ug386.pdf](http://www.xilinx.com/support/documentation/user_guides/ug386.pdf).
  - [53] Texas Instruments. *TPS62510: 1.5A Low Vin High Efficiency Step-Down Converter (Rev. A)*. 2009. URL: <http://www.ti.com/product/tps62510>.
  - [54] Texas Instruments. *TPS54319: 2.95-V to 6-V Input, 3-A Output, 2-MHz, Synchronous Step-Down Switcher With Integrated FETs*. 2010. URL: <http://www.ti.com/product/tps54319>.
  - [55] Texas Instruments (former National Semiconductor). *LMK01000 - Active 1.6GHz High-performance Clock Buffer, Divider, and Distributor*. 2009. URL: <http://www.ti.com/product/lmk01010>.
  - [56] Xilinx. *LogiCORE IP Aurora 8B/10B v8.1*. 2012. URL: [http://www.xilinx.com/support/documentation/ip\\_documentation/aurora\\_8b10b/v8\\_1/aurora\\_8b10b\\_ds797.pdf](http://www.xilinx.com/support/documentation/ip_documentation/aurora_8b10b/v8_1/aurora_8b10b_ds797.pdf).



**Danksagung** Das Mitwirken am DSSC Projekt war für mich eine sehr interessante und lehrreiche Erfahrung. Besonders freue ich mich aber über die Bekanntschaften vieler neuer und vor allem netter Menschen, von denen ich *Dr. Ing. Matteo Porro*, *Dr. Chris Youngman*, *Dr. Ing. Karsten Hansen*, *Helmut Klär* und *Prof. Dr. Peter Fischer* stellvertretend nennen möchte.

Besonderer Dank gilt meinem Doktorvater Herrn *Prof. Dr. Reinhard Männer*. Sie haben mir durch Ihr entgegengebrachtes Vertrauen und Ihre Überzeugung das Mitwirken an diesem spannenden Projekt überhaupt erst ermöglicht.

Ebenso möchte ich meinem Betreuer *Dr. Andreas Kugel* danken. Die Zusammenarbeit mit Ihnen hat mich sehr viel gelehrt, und Sie waren mir immer mit hilfreichem Rat zur Seite gestanden. Für das stets aufheiternde Arbeitsklima danke ich meinen Bürokollegen *Nicolai Schroer*, *Moritz Kretz* und insbesondere *Andreas Wurz*. Eure Ratschläge und Sprüche waren ebenso kompetent wie amüsant.

Weiter möchte ich meinen Bandkollegen und Freunden von *RockPaperScissors* und *Kristina Neureuther & Band* für das entgegengebrachte Verständnis bedanken. Gerade in den letzten Monaten habt Ihr oft auf mich verzichten müssen.

Außerdem seien an dieser Stelle *Julia Weinmann-Klausmann* und *Rouven Klausmann* erwähnt. Vielen Dank für Eure Hilfe und Geduld beim finalen Druckprozess.

Ganz besonders danke ich meiner Freundin *Corinna Pfisterer*. Nicht nur, weil Du mich immer wieder mit Deinen kulinarischen Kreationen freudig überrascht hast. Sondern vor allem, weil Du mir in den vergangenen Monaten sehr viel gegeben hast und dabei selbst auf vieles verzichten musstest. Dafür möchte ich Dir von ganzem Herzen danken.

Zuletzt gilt ein großer Dank meiner Familie *Heinrich* und *Waltraud Gerlach* sowie *Matthias Gerlach*. Ihr wart immer an meiner Seite und habt mir geholfen, meine Ziele weiter zu verfolgen und sie zu erreichen.

...as my guitar lies bleeding in my arms!  
*My Guitar Lies Bleeding in My Arms (Bon Jovi, 1995, "These Days")*



# Eidesstattliche Versicherung gemäß §7, Absatz 2, Buchstabe c) der Promotionsordnung der Universität Mannheim zur Erlangung des Doktorgrades der Naturwissenschaften

1. Bei der eingereichten Dissertation zum Thema

*Development of the DAQ Front-end for the DSSC Detector at the European XFEL*

handelt es sich um mein eigenständig erstelltes eigenes Werk.

2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtliche Zitate aus anderen Werken als solche kenntlich gemacht.
3. Die Arbeit oder Teile davon habe ich ~~wie folgt~~ / bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.

Titel der Arbeit:

.....

Abschluss:

.....

4. Die Richtigkeit der vorstehenden Erklärung bestätige ich.
5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt.

Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Mannheim, im März 2013

*Thomas Gerlach*